



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JOEL NIEMINEN

TOIMINNANOHJAUSJÄRJESTELMÄN VALINTA AVOIMEN
LÄHDEKOODIN NÄKÖKULMASTA

Diplomityö

Tarkastaja: professori Hannu Jaakkola
Tarkastaja ja aihe hyväksytty
Tieto- ja sähkötekniikan
tiedekuntaneuvoston kokouksessa
14. elokuuta 2013

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

NIEMINEN, JOEL: Toiminnanohjausjärjestelmän valinta avoimen lähdekoodin näkökulmasta

Diplomityö, 62 sivua

Syyskuu 2013

Pääaine: Ohjelmistotekniikka

Tarkastaja: professori Hannu Jaakkola

Avainsanat: Toiminnanohjausjärjestelmä, avoin lähdekoodi, implementointi

Työssä käydään läpi toiminnanohjausjärjestelmän (ERP) implementointia ja asiaa tarkastellaan myös avoimen lähdekoodin näkökulmasta. Tavoitteena on luoda katsaus aiheeseen liittyviin asioihin, kuten ERP-järjestelmän vaatimusmäärittelyyn ja projektin onnistumisen edellytyksiin. Työssä selvitetään myös avoimen lähdekoodin ERP-järjestelmän valintakriteereitä, luodaan katsaus avoimen lähdekoodin ohjelmistoihin ja ERP-järjestelmän tuotteisiin. Työ on kirjallisuuskatsaus.

Toiminnanohjausjärjestelmän hankinta on suuri investointi ja iso askel yritykselle, oli sitten kysymyksessä avoimen lähdekoodin tai suljetun toiminnanohjausjärjestelmän valinta. Sen implementointi saattaa kaataa koko yrityksen epäonnistuessaan. Tästä johtuen yrityksen kannattaa perehtyä aiheeseen hyvin. Yrityksellä tulee olla käytössään tarvittavat resurssit, avainhenkilöt ja osaaminen, jotta varmistetaan onnistuminen. ERP-järjestelmän konsultteja tulee käyttää, koska heillä hyvää tietotaitoa. Tulee myös varmistaa, että konsultin tietotaitoa siirtyy yrityksen henkilöille mahdollisimman paljon ja dokumentaation taso on hyvä, jolloin saadaan sijoitukselle vastinetta. Koulutukseen tulee panostaa; koska jos järjestelmää ei osata hyödyntää, sen käyttö jää helposti vajaaksi.

Seuraavia asioita voidaan nostaa esiin, kun yritys miettii avoimen lähdekoodin ERP-järjestelmän hankintaa. Jos yrityksellä on käytössään omaa henkilöstöä, jotka ovat kiinnostuneita ja riittävän osaamisen omaavia, niin avoimen lähdekoodin ERP-järjestelmän hankinta saattaa tulla kysymykseen. On huomattavaa, että avoimen lähdekoodin ERP-järjestelmää ei kannata valita pelkästään sen takia, että sitä voidaan sanoa ilmaiseksi, vaikkakin se on jo hyvä syy itsessään. Avoimen lähdekoodin ERP-järjestelmän implementointiin kuuluu muun muassa konsultointia, ylläpitoa ja koulutusta, joista muodostuu kustannuksia. Hyvänä puolena on, että avoimen lähdekoodin ERP-järjestelmän kehitystyöhön ja testaukseen osallistuu paljon henkilöitä eri puolilta maailmaa, joten se ei ole rajoittunut siihen, että toimittaja lopettaa toiminnan. Varteenotettava mahdollisuus on vain ladata internetistä soveltuva avoimen lähdekoodin ERP-järjestelmä ilmaiseksi ja asettaa tähän sopivat resurssit käyttöön.

ABSTRACT

TAMPEREUNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

NIEMINEN, JOEL: ERP system selection of Open Source perspective

Master of Science Thesis, 62 pages

September 2013

Major: Software Engineering

Examiner: Professor Hannu Jaakkola

Keywords: Enterprise Resource Planning, Open Source, ERP, implementation

This thesis examines the implementation of Enterprise Resource Planning (ERP), also from the open source viewpoint. The objective of the thesis was to provide a comprehensive report on the issues involved, such as the requirement specifications of ERP systems, prerequisites for project success, as well as selection criteria for an open source ERP system, while taking a look at different kinds of open source software and ERP system products. The thesis is a review of existing literature.

The Enterprise Resource Planning system implementation is a large investment and a big step for the company, whether it relates to an open source or a closed ERP system selection. Implementation failures could ruin the whole company, which is why the company should explore the topic well ahead of time. The company must have the necessary resources and know-how as well as the right people in order to ensure success. ERP consultants should be used, because they have the required know-how. It should also be ensured that as much as possible of the consultant's know-how is relayed to the company's employees and that sufficient documentation is maintained in order to provide value for investment. Training investments must be made because if people are not able to use the system, it is easy to under-utilise it.

The following items can be considered when the company thinks about making an open source ERP system acquisition. If the company's own staff possesses a sufficient level of interest and know-how, it may be relevant to acquire an open source ERP system. It should be noted that the open source ERP system should not be chosen only because it is free, even though it is a valid reason in itself. Among other things, the open source ERP system implementation includes consulting, maintenance and training, which create costs. A positive thing is that a lot of people across the globe participate in the development and testing of open source ERP systems, so it is not limited to the supplier ceasing operations. One worthy option is to just download the appropriate open source ERP system for free from the Internet and follow it up by utilising relevant resources.

ALKUSANAT

Kiitokset ohjaajilleni professori Marjan Herickolle ja professori Hannu Jaakkolalle heidän tuestaan työtäni kohtaan. Erityiskiitos myös professori Tatjana Welzerille, joka tuki opintojani Sloveniassa.

Pori 15.08.2013

Joel Nieminen

SISÄLLYS

Tiivistelmä.....	ii
Abstract.....	iii
1 Johdanto.....	1
2 ERP-järjestelmän historia.....	3
2.1 ERP-järjestelmien kehittyminen.....	3
2.2 ERP-järjestelmän arkkitehtuuri.....	7
3 Yleistä ERP-järjestelmästä.....	9
3.1 ERP-järjestelmän määritelmiä eri lähteistä.....	9
3.2 ERP-järjestelmän tärkeimmät moduulit.....	10
4 ERP-järjestelmän elinkaari.....	12
4.1 ERP-järjestelmän elinkaaren vaiheet.....	12
4.2 ERP-järjestelmän elinkaaren näkökulmat.....	14
5 ERP-järjestelmän vaatimusmäärittely pk-yrityksessä.....	15
5.1 Vaatimusmäärittely ERP-järjestelmän lähtökohtana.....	15
5.2 Vaatimusmäärittelyn näkökulmat.....	16
5.3 Loppukäyttäjän näkökulma toiminnanohjausjärjestelmän vaatimuksiin.....	17
5.4 Järjestelmätyypit ja niiden vaikutus vaatimusmäärittelyyn.....	18
5.5 Vaatimusmäärittelyn vaiheet.....	20
5.6 Yhteenveto vaatimusmäärittelystä.....	22
6 ERP-projektin onnistumisen edellytykset.....	23
7 Avoimen lähdekoodin ohjelmisto.....	29
7.1 Avoimen lähdekoodin määritelmä.....	30
7.2 Ohjelmistolisenssit.....	31
7.3 Avoimen lähdekoodin ohjelmistojen historiaa.....	34
7.4 Ohjelmistokehitys.....	35
7.5 Liiketoiminnan näkökulma.....	37
7.6 Avoimen lähdekoodin ohjelmistot vs. suljetut ohjelmistot.....	38
7.7 Avoimen lähdekoodin ohjelmistojen tulevaisuus.....	41
8 ERP-järjestelmän valintakriteerit avoimen lähdekoodin näkökulmasta.....	42
9 Avoimen lähdekoodin ERP-järjestelmiä.....	48
10 Johtopäätökset.....	61
Lähteet.....	63

1 JOHDANTO

Tänä päivänä yritykset joutuvat toimimaan kovassa kilpailutilanteessa. Yritysten pitää ennakoida, vastata ja reagoida markkinoiden kasvaviin vaatimuksiin. Yrityksellä pitää olla hyvä liiketoimintastrategia, joka tehokkaasti hyödyntää tietotekniikan käyttöä. Yksi yleisimmistä ratkaisuista yrityksillä on käyttää toiminnanohjausjärjestelmää.

Toiminnanohjausjärjestelmä on laaja ohjelmistokokonaisuus, joka koostuu useista erillisistä moduuleista. Se tarjoaa yritykselle työkaluja päätöksentekoon, myynnin koordinointiin, markkinointiin, logistiikkaan, ostoon, budjetointiin, tuotekehitykseen ja henkilöstöhallintaan. Toiminnanohjausjärjestelmä tukee ennen kaikkea yritysten tarjous-myynti-tilaus-toimitusprosesseja. Toiminnanohjausjärjestelmään saadaan muun muassa linkitettyä verkkoliiketoiminta ja toimitusketjun hallinta. Yhdistämällä toimitusketjut muiden yritysten järjestelmiin läpimenoaika saadaan lyhemmäksi ja varastot pidettynä pieninä. Toiminnanohjausjärjestelmä on kallis ja iso investointi yritykselle.

Avoimen lähdekoodin periaate tarkoittaa yleisesti, että ohjelman lähdekoodi on vapaasti käytettävissä. Avoimen lähdekoodin ohjelmien jakelu on yleisesti ottaen vapaata, ainoastaan palvelut ja lisäominaisuudet ovat maksullisia, esimerkiksi räätälöinti, koulutus ja tekninen tuki. Avoimen lähdekoodin ohjelmistot ovat alkaneet syrjäyttämään suljettuja, kaupallisia ohjelmistoja. Avoimen lähdekoodin etuja käyttäjien mielestä ovat muun muassa suurempi vakaus, korkea turvallisuuden taso, lisenssimaksujen pienuus, lähdekoodin muokattavuus, mahdollisuus kehittyä paremmaksi ohjelmoijaksi ja riippumattomuus suurista ohjelmistoyrityksistä. Avoimen lähdekoodin ohjelmiston kehitys ja testaus on nopeaa, koska ympäri maailman on testaajia ja kehittäjiä runsaasti. Avoimen lähdekoodin ohjelmistojen merkitys ohjelmistotuotannossa on voimakkaassa kasvussa ja tätä ilmiötä on viime aikoina tutkittu paljon.

Avoimen lähdekoodin toiminnanohjausjärjestelmiä on saatavilla nykyään hyvin suuri määrä. Nykyään niitä käyttävät sekä pienet että suuret yritykset ympäri maailman. Avoimen lähdekoodin toiminnanohjausjärjestelmät kiinnostavat yrityksiä tänä päivänä enemmän kuin koskaan. Voidaan nostaa seuraavat kolme merkittävää etua esiin, jotka tukevat avoimen lähdekoodin toiminnanohjausjärjestelmän valintaa:

- Lähdekoodin vapaa saatavuus tekee räätälöinnin helpoksi, varsinkin kun järjestelmä todennäköisesti pitää joka tapauksessa räätälöidä yrityksen liiketoimintaprosesseihin;
- Kaupallista toiminnanohjausjärjestelmää käyttävä yritys on riippuvainen tuotteen valmistajasta ja jakelijoista, jos toinen tai molemmat katoavat, päivitykset ja ylläpito aiheuttavat merkittäviä ongelmia;
- Kaupallisen toiminnanohjausjärjestelmän lisenssi on hyvin kallis, kun taas avoimen lähdekoodin lisenssi on ilmainen.

Tämän tutkimuksen tarkoituksena herättää kiinnostus avoimen lähdekoodin toiminnanohjausjärjestelmiin. Työ on kirjallisuuskatsaus. Työ on tehty, koska aiheesta löytyy suomalaista kirjallisuutta suhteellisen vähän, kun taas englannin kielistä materiaalia on tarjolla runsaasti. Tarkoitus on esittää asioita, jotka pitää ottaa huomioon toiminnanohjausjärjestelmän hankkimisessa ja toteuttamisessa eli implementoinnissa, ottaen huomioon avoimen lähdekoodin näkökulma. Toiminnanohjausjärjestelmän implementointi on iso askel ja suuri investointi yritykselle, oli sitten kysymyksessä avoimen lähdekoodin tai suljetun toiminnanohjausjärjestelmän valinta. Tästä johtuen yrityksen kannattaa perehtyä asiaan hyvin, jotta toiminnanohjausjärjestelmän implementointi saadaan onnistuneesti läpi.

Seuraavien lukujen sisältö lyhyesti: Luvussa kaksi käydään läpi toiminnanohjausjärjestelmän historiaa; Kolmannessa luvussa selvitetään, mikä toiminnanohjausjärjestelmä on ja käydään läpi toiminnanohjausjärjestelmän moduuleita; Luvussa neljä esitetään toiminnanohjausjärjestelmän elinkaari ja tarkastellaan sitä eri näkökulmista; Luvussa viisi luodaan katsaus toiminnanohjausjärjestelmän hankintaan liittyvästä vaativuusmäärittelystä pk-yrityksen näkökulma huomioon ottaen; Kuudennessa luvussa avataan projektinäkökulmaa toiminnanohjausjärjestelmän hankintaan liittyen; Seitsemännessä luvussa käydään avoimen lähdekoodin ohjelmistoon liittyviä asioita läpi kattavasti; Luvussa kahdeksan käydään läpi toiminnanohjausjärjestelmän valintakriteereitä avoimen lähdekoodin näkökulmasta; Luvussa yhdeksän luodaan lyhyt katsaus avoimen lähdekoodin ERP-järjestelmiin; Luvussa kymmenen esitetään yhteenveto työstä.

2 ERP-JÄRJESTELMÄN HISTORIA

2.1 ERP-järjestelmien kehittyminen

Toiminnanohjausjärjestelmästä yleisesti käytetty englanninkielinen lyhenne (ERP), tulee sanoista Enterprise Resource Planning. Toiminnanohjausjärjestelmän historia alkaa jo ennen tietojärjestelmien tuloa ja se seuraa melko hyvin tietokoneen kehitystä. Tilauksia ja varaston hallintaa hoidettiin pitkälti manuaalisesti. Varastossa pidettiin riittävästi tavaraa, jotta tilaukset saatiin hoidettua. Varaston seurantaan aloitettiin kehittää ohjelmistoja 1960-luvulla. Ne olivat varsin yksinkertaisia, yritykselle kehitettyjä järjestelmiä. Niitä käytettiin lähinnä varaston seurantaan.

Materiaalinhallinta perustui lähinnä tilauksiin ja oletukseen tuotteiden pitkistä elinkaarista. Tarkoituksena oli varastoida tavaraa varastoon asiakkaan kysyntää seuraten ja hieman enemmän kuin tarve oli. ERP-järjestelmän historia on ollut erilainen maasta tai maanosasta riippuen, maan tietotekninen kehitys on myös vaikuttanut asiaan.

Tuotantoa tukevien järjestelmien kehitys alkoi 1970-luvulla. Valmistava teollisuus otti materiaalien vaatimuksen suunnittelun käyttöön ja siitä tuli ensisijainen valmistuksen toimintatapa. Ohjelmistot suorittivat materiaalien tarvelaskentaa. Materiaalien vaatimuksen suunnittelun englanninkielinen lyhenne (MRP), tulee sanoista Material Requirements Planning.

Tässä vaiheessa osto hyödynsi osaluetteloa osien tilauksessa (BOM), joka on lyhenne englanninkielisen sanoista Bill Of Material. Osien kehitys ja hallinta tapahtui myös osaluettelon puitteissa. Tämä konsepti laajeni kattavuudeltaan tilauksesta varaston materiaalin hallintaan tehtaassa. Se laajeni myös henkilöstön- ja toimituksen suunnitteluun. Tästä seurasi valmistuksen vaatimusten suunnittelu. Valmistuksen vaatimusten suunnittelun englanninkielinen nimi on Manufacturing Resource Planning (MRP II). Valmistuksen vaatimusten suunnitteluun sulautettiin rahoitus ja kirjanpito, henkilöstöhallinnan toiminnat, jakelutoiminnat ja johtamisen hallinta. Nämä tulivat kokonaisvaltaisesti kattamaan koko yrityksen kaikki osa-alueet ja tätä lopulta kutsuttiin toiminnanohjausjärjestelmäksi (ERP). Seuraavassa käydään näitä läpi hieman yksityiskohtaisemmin.

[1; 2; 3]

Varastonhallinta

Varastonhallinta oli esiaste toiminnanohjausjärjestelmissä. Tämä tapahtui 1960-luvulla. Varastonhallintaa hallittiin osaluettelolla ja tuotannon aikataululla (MPS), joka on lyhenne englanninkielien sanoista Master Production Schedule. Se on suunnitelma yksittäisten hyödykkeiden tuottamiseen kussakin ajassa. Varastonhallinta oli yhdistelmä tietotekniikkaa ja liiketoimintaprosessien ohjailua, tarkoituksena pitää varasto oikealla tasolla.

Varaston aktiviteettien hallinta tunnistaa seuraavia varaston täydennyksen vaatimuksia: tavoitteiden asettelu, täydentämisen tekniikoiden ja vaihtoehtojen perustaminen, nimikkeiden käytön seuranta, nimikkeiden määrän sovittaminen oikealle tasolle ja varaston tilan raportointi.

Materiaalien vaatimusten suunnittelu

Materiaalien vaatimusten suunnittelua (MRP) alettiin käyttää 1960-luvulla ja se vakiinnutti asemansa 1970-luvulla. MRP-järjestelmä oli seurausta osaluettelon käytöstä. Valmistuksen ja tuotannon suunnittelijat hakivat parempia ja tehokkaampia toimintatapoja tilata materiaaleja ja komponentteja. He havaitsivat, että MRP-järjestelmä oli täydellinen vastaus heidän ongelmiinsa.

Materiaalien vaatimusten suunnittelu järjestelmä vastaa seuraaviin kysymyksiin:

- Mitä tuotteita aiotaan valmistaa?
- Mitä materiaaleja tarvitaan näihin tuotteisiin?
- Mitä materiaaleja on varastossa?
- Mitä tavaroita pitää ostaa?

MRP-järjestelmä käyttää tuotantoaikataulua selvittääkseen vastauksen ensimmäiseen kysymykseen, mitä tuotteita aiotaan valmistaa. Se antaa yksityiskohtaisesti materiaalien vaatimukset valmistettaviin tuotteisiin osaluettelosta. Se antaa tiedot varastosta. Tämän jälkeen MRP-järjestelmä antaa nimikkeet, joita pitää ostaa, jotta saadaan valmistettua halutut tuotteet. Täten MRP-järjestelmä ratkaisee isoja valmistuksen ja tuotannon suunnittelijoiden ongelmia ja tekee tuotteiden valmistamisen helpommaksi.

Materiaalin vaatimusten suunnittelu hyödyntää ohjelmistoja aikatauluttaakseen tuotannon prosesseja. MRP-järjestelmä generoi aikataulun operaatioille. Materiaalin osto pohjautuu tuotannon vaatimukseen lopputuotteesta, tuotantojärjestelmän rakenteeseen ja varaston tasoon. MRP-järjestelmässä on paljon parametreja, joilla hallitaan kyseistä toimintaa. Tuotantoaikataulu vahvistetaan lopuksi, kun materiaalien saatavuus on varmistettu.

Tavoitteena oli pitää varastomäärät sopivina ja varmistaa, että jokaisessa työvaiheessa oli riittävästi resursseja saatavina. Toiminnallisuuteen kuului myös taloudellisen eräkoon selvittäminen tuotantoa varten; tämä ominaisuus oli kehittyneemmissä järjestelmissä. Tuotantoaikataulu lasketaan saatujen tilausten ja myyntiennusteen mukaan.

Takaisinkytketty MRP

Tämä oli toiminnanohjausjärjestelmän kehityksessä välivaihe, joka lisäsi ominaisuuksia materiaalien vaatimusten suunnittelu järjestelmään. Nämä MRP-järjestelmän kehittäjät ymmärsivät, että siinä oli enemmän kapasiteettia, kuin tuottaa materiaaleja uudestaan suunnitellussa aikataulussa. MRP-järjestelmään on syötetty tilausten perusteella nimikkeiden toimituspäivät. MRP-järjestelmä havaitsee ja varoittaa, jos nimikkeet eivät saavu toimituspäivään mennessä. Tämä uusi ominaisuus helpotti vaikeasti saatavien osien seuranta valmistusprosessissa.

Pian suunnittelutekniikat kapasiteetin vaatimuksista sulautettiin MRP-järjestelmään. Työkalut oli kehitetty tukemaan myynnin suunnittelua ja tuotannon tasoa, parantamaan tuotannon aikataulua, ennakoimiseen, myynnin suunnitteluun, kapasiteetin suunnitteluun ja prosessien ohjaamiseen. Tuotanto ja toimittajien aikataulutekniikat automatisoivat prosessia organisaation sisä- ja ulkopuolella. Nämä toiminnot lisättiin MRP-järjestelmään. Näistä kehityksen tuloksena luotiin takaisinkytketty MRP-järjestelmä, jonka englanninkielinen nimi on Closed-Loop MRP.

Takaisinkytketty MRP-järjestelmä ei ollut tarkoitettu ainoastaan materiaalien vaatimuksien suunnitteluun, vaan se oli myös toimintojen sarja automatisoimaan koko tuotantoprosessia. Se sisältää työkaluja ja tekniikoita käsitellä sekä prioriteetti että kapasiteetti. Takaisinkytketty MRP-järjestelmä tukee suunnittelua ja tuotantoa. Se tarjoaa palautteen tuotannosta takaisin suunnitteluun. Täten se mahdollistaa revisioidin ja päivityksen suunnittelussa, ottaen huomioon tämänhetkisen tuotannon tai prioriteettien muutokset.

Valmistuksen vaatimusten suunnittelu

Kolmas askel ERP-järjestelmän kehityksessä oli valmistuksen vaatimusten suunnittelu järjestelmä (MRP II), jonka englanninkielinen nimi on Manufacturing Resource Planning. Tämä tapahtui 1980-luvulla. Se oli seuraava looginen askel takaisinkytketyn MRP-järjestelmän jälkeen. Se sisältää seuraavat lisäominaisuudet:

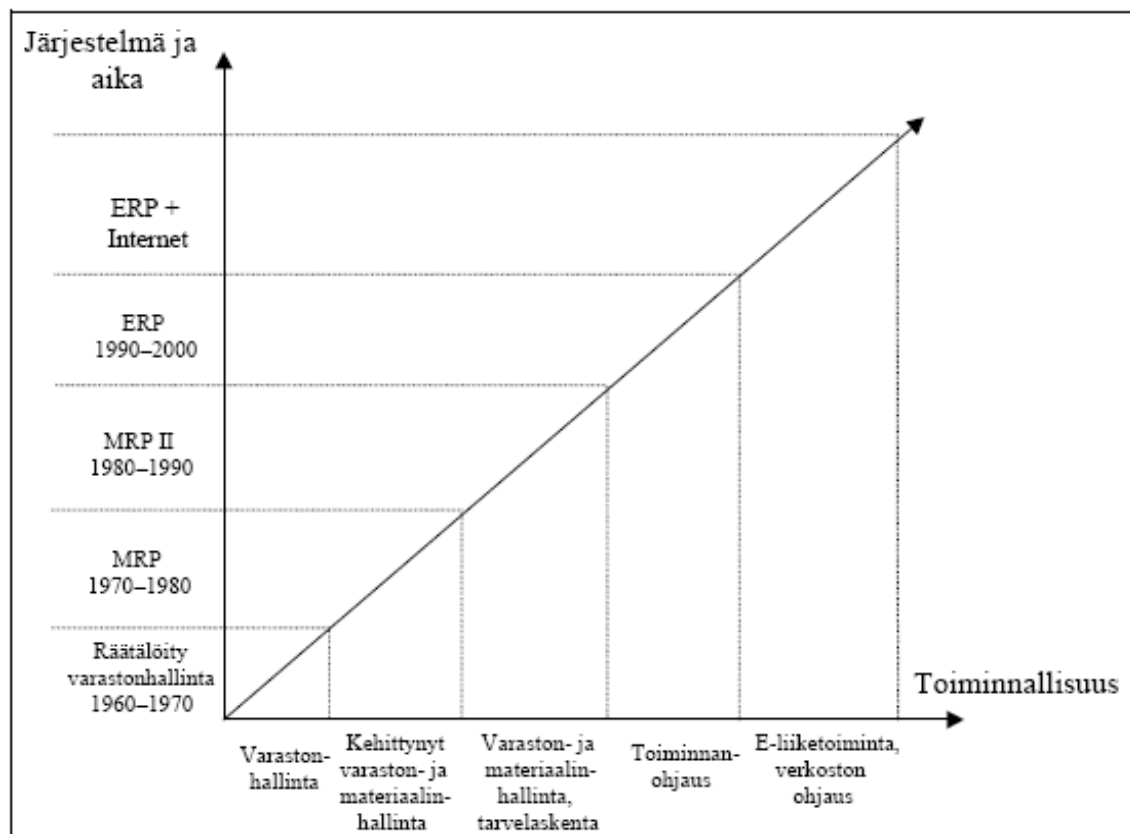
- Myynnin ja operationaalisen suunnittelun;
- Liiketoiminnallisen liitännän;
- Simulaatio-ominaisuuden, joka auttaa henkilöstöä päätöksen teossa.

MRP II-järjestelmä on menetelmä tehokkaaseen suunnitteluun, jossa kaikki resurssit otetaan huomioon organisaatiossa. Se käsittelee operatiivisen yksikön resurssit ja liiketoiminnan suunnittelun. Simulaatio-toiminnolla voidaan vertailla etukäteen eri mahdollisuuksia. Valmistuksen vaatimusten suunnittelu-järjestelmä muodostuu useista toiminnoista, jossa kaikki toiminnot on linkitetty yhteen. Niitä ovat liiketoimintojen suunnittelu, myynti ja operaatioiden suunnittelu, menekin hallinta, tuotannon suunnittelu, aikataulutus ja materiaalin vaatimusten suunnittelu. Näiden tehtävänä on tukea tuotantoa kapasiteetin ja materiaalin hallinnassa. MRP II-järjestelmästä saadaan myös monenlaisia raportteja helpottamaan yrityksen toimintaa, esimerkiksi osto-, toimitus- ja varastoraportit.

ERP-järjestelmä

Viimeinen askel kehityksessä oli varsinainen toiminnanohjausjärjestelmä (ERP). Tämä tapahtui 1990-luvulla. ERP-järjestelmä on seuraus ja laajennus MRP-järjestelmästä ja sisältää myös kaikki MRP II-järjestelmän ominaisuudet.

Kuvassa 1 on kuvattu ERP-järjestelmän historiaa.



Kuva 1. ERP-järjestelmien kehityspolku.

Liiketoiminnan näkökulmasta ERP-järjestelmä on laajentunut valmistuksen prosessien koordinoinnista yrityksen toimintojen laaja-alaiseen integroitiin koko organisaatiossa. Teknologisesta näkökulmasta ERP-järjestelmä on kehittynyt yksinkertaisesta toteutuksesta enemmän joustavaan asiakas-palvelin kerrosarkkitehtuuriin. [2; 3]

2.2 ERP-järjestelmän arkkitehtuuri

ERP-järjestelmän toimittajat, joilla oli kokemusta MRP-järjestelmästä ja liiketoiminnan ohjelmista, ymmärsivät rajoitukset vanhoissa tietojärjestelmissä. Näitä käytettiin suurissa yrityksissä 1970- ja 1980-luvuilla. Osa näistä vanhoista järjestelmistä kehitettiin yleensä tietylle yritykselle. Yrityksissä oli käytössä eri toimittajien ohjelmia, näissä oli useasti käytössä erilaisia tietokantoja ja ohjelmointikieliä ja ne eivät olleet keskenään yhteensopivia. Järjestelmät eivät tukeneet saumatonta tiedonkulkua eri järjestelmien välillä. Tällaisissa järjestelmissä oli myös vaikeaa lisätä kapasiteettia. Käyttäjät eivät voineet päivittää niitä organisaation liiketoiminnan muutoksiin, strategiaan tavoitteisiin ja uuteen tietotekniikkaan.

ERP-järjestelmältä vaaditaan seuraavia toiminnallisia ominaisuuksia:

- Modulaarinen rakenne käsittäen monia erilaisia liiketoiminta moduuleja, kuten rahoitus, valmistus, kirjanpito ja jakelu;
- Moduulit ovat integroitua ja ne tarjoavat saumattoman tietovirran eri moduulien välillä lisäten toiminnan avoimuutta vakioliitännöillä;
- Moduulit toimivat reaaliajassa verkossa ja niissä on eräajo valmiudet;
- Järjestelmä käyttää yhteistä keskitettyä tietokannan hallintajärjestelmää;
- Ne ovat yleensä monimutkaisia ja kalliita järjestelmiä;
- Ne ovat joustavia ja tarjoavat parhaita liiketoimintamalleja;
- Järjestelmän konfigurointi ja sovittaminen yrityksen liiketoiminnan prosesseihin on aikaa vievä toimenpide;
- Ne ovat tai tulevat olemaan internet-tietoverkossa.

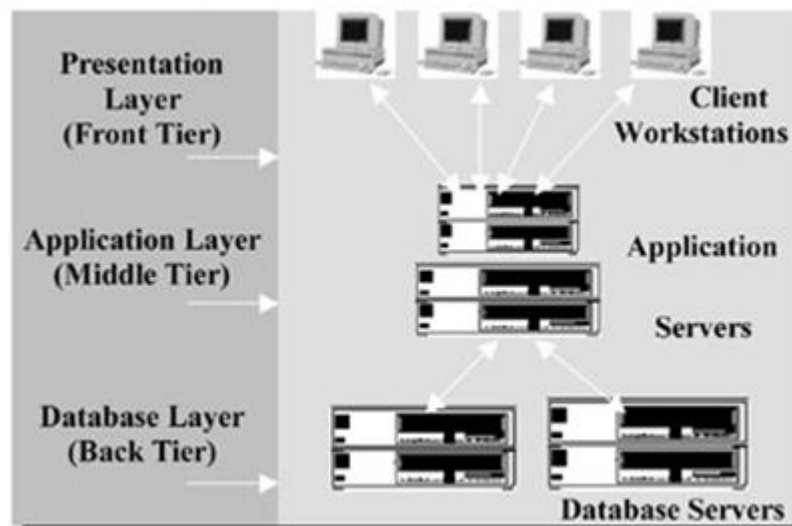
Eri ERP-järjestelmien toimittajat tarjoavat järjestelmiensä kanssa lisäominaisuuksia, mutta ydinmoduulit ovat lähes kaikille samat. ERP-järjestelmän moduulit toimivat joko erillisenä yksikkönä tai useita moduuleja voidaan yhdistää muodostaen näin yhtenäisen järjestelmän. Moduuleita käydään läpi luvussa 3.

ERP-järjestelmät käyttävät asiakas-palvelin mallia luomaan hajautetun tietojenkäsittely-ympäristön. Asiakas-palvelin mallissa operoidaan useiden loppukäyttäjien päätelaitteiden, kuten pöytätietokoneiden kanssa. Päätelaite lähettää pyynnön sovelluspalvelimelle, joka puolestaan saa pyydettyyn palveluun liittyvät tiedot tietokantapalvelimelta. Pyyntö voi olla yksinkertaisia tiedostoja, data-arvoja, viestintäpalveluita, tapahtumien käsittelyä tai päivityksiä tiedostoon. Toimiakseen ERP-

järjestelmä tarvitsee suhteellisen tehokkaan tietokoneen ja erittäin tehokkaan palvelimen, koska se suorittaa suuren määrän tapahtumia.

Yleisesti on käytössä kolmitasoinen kerrosarkkitehtuuri, kuten kuvassa 2 on esitetty:

- Käyttöliittymäkerros (Presentation Layer): Graafinen käyttöliittymä, tietojen syöttäminen selaimella tai pääsy järjestelmän toimintoihin;
- Sovelluskerros (Application Layer): Liiketoimintasäännöt, toiminnot, logiikka, ohjelmat vastaanottavat/lähehtävät tiedot tietokanta palvelimelta/palvelimelle;
- Tietokantakerros (Database Layer): Organisaation toimintojen hallintaan tai tapahtumatiedot mukaan lukien metatiedot.



Kuva 2. ERP-järjestelmän kolmitasoinen kerrosarkkitehtuuri.

Tämä looginen järjestely toimii seuraavasti: ERP-järjestelmän käyttäjät toimivat käyttöliittymätasolla, prosessointimoduulit toimivat keskikerroksen sovelluspalvelimissa ja tietokantajärjestelmä toimii tietokantapalvelimissa.

[4]

3 YLEISTÄ ERP-JÄRJESTELMÄSTÄ

3.1 ERP-järjestelmän määritelmiä eri lähteistä

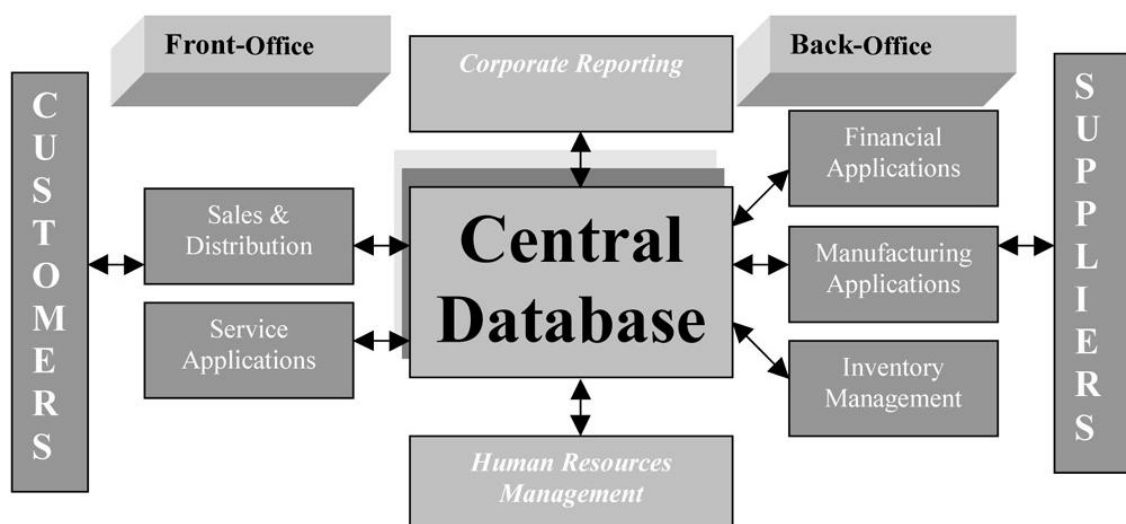
Davenportin mukaan ERP-järjestelmä koostuu kaupallisesta ohjelmistopakettista, joka lupaa saumattoman integraation koko yrityksen informaatiosta, joka kulkee läpi yrityksen, sisältäen seuraavat osa-alueet: rahoitus, kirjanpito, henkilöstöresurssit, toimitusketju ja asiakkaan tiedot. [5]

ERP-järjestelmät ovat konfiguroitavia tietojärjestelmien paketteja, jotka integroivat yrityksen koko informaatio pohjaiset prosessit, jotka sisältävät kaikki toiminnalliset alueet organisaatiossa. [6]

Yksi tietokanta, yksi sovellus ja yhtenäinen käyttöliittymä koko yritykseen. [7]

ERP-järjestelmät ovat tietokonepohjaisia järjestelmiä, jotka on suunniteltu käsittelemään ja ohjaamaan yrityksen liiketoimia, helpottamaan integraatiota, reaaliaikaista suunnittelua ja tuotantoa, ja jotka pohjautuvat asiakkaiden tilauksiin. [8]

ERP-järjestelmää voidaan havainnollistaa kuvan 3 mukaisesti.



Kuva 3. ERP-järjestelmä. [5]

ERP-järjestelmä on laaja ohjelmistokokonaisuus, joka koostuu useista erillisistä moduuleista. Sen tarkoituksena on ennakoida ja tasapainottaa kysyntää. Se tarjoaa

ennakoimiseen, suunnitteluun ja aikataulutukseen työkaluja, jotka on linkitetty asiakkaitten ja toimittajien toimitusketjuun. Se tarjoaa työntekijöille työkaluja päätöksentekoon, myynnin koordinointiin, markkinointiin, logistiikkaan, ostoon, budjetointiin, tuotekehitykseen ja henkilöstöhallintaan. [3]

Toiminnanohjausjärjestelmän määritelmä on laajentunut vuosien kuluessa ja riippuu paljon siitä, mistä näkökulmasta asiaa tarkastellaan. Esimerkiksi liikkeenjohto tarkastelee asiaa laajasta näkökulmasta. Toiminnanohjausjärjestelmät tukevat ennen kaikkea yritysten tarjous-myynti-tilaus-toimitusprosesseja. ERP-järjestelmällä ei tarkoiteta pelkästään yrityksen käytössä olevaa järjestelmää, vaan johtamisen näkökulmasta se tarkoittaa tiettyjen prosessien ohjauksessa käytettäviä periaatteita ja menetelmiä sekä niitä tukevaa tietotekniikkaa. ERP-järjestelmään voi kuulua useita eri ohjelmistoja. [9]

Ohjelmiston arkkitehtuuri helpottaa integrointia moduulien välillä tarjoten informaatiovirran yrityksen kaikkiin toimintoihin johdonmukaisella tavalla. ERP-järjestelmän avulla yritykset voivat toteuttaa yhden integroidun järjestelmän korvaamalla tai uudistamalla vanhoja tietojärjestelmiä, jotka ovat usein ristiriidassa keskenään. [4]

3.2 ERP-järjestelmän tärkeimmät moduulit

ERP-järjestelmään voi kuulua seuraavanlaisia moduuleita: kirjanpito, taloushallinto, valmistuksen hallinta, tuotannon hallinta, logistiikan hallinta, myynnin ja toimituksen hallinta, henkilöstöjohtaminen, toimitusketjun hallinta, asiakassuhteiden hallinta, internet-liiketoiminta, tuotetiedonhallinta. Moduuleiden nimet saattavat vaihdella lähteistä riippuen. Seuraavassa tarkastellaan muutamia moduuleita tarkemmin.

Tuotetiedonhallinta

Tuotetiedonhallintajärjestelmä sisältää tiedot tuotteesta. Sillä hallitaan koko tuotteen elinkaari. Tuotetieto sisältää kaiken informaation, joka tarvitaan asiakkaalle toimitettavan tuotteen konseptoimiseen, suunnitteluun, hankintaan, valmistamiseen, myymiseen, huoltamiseen ja hävittämiseen. ERP-järjestelmissä ei ole niin kattava tuotetiedonhallinta, kuin aidossa tuotetiedonhallintajärjestelmässä. Joissakin ERP-järjestelmissä sitä ei ole ollenkaan. Esimerkiksi konepajateollisuudessa tuotetiedonhallinta on hyvä olla olemassa; ilman tätä ominaisuutta tuotetiedonhallinta on haastavaa. Myös erillisen liitännän mahdollisuus tuotetiedon hallintaan järjestelmään kannattaa ottaa huomioon.

Tuotannonohjaus

Tuotannonohjauksen tavoitteena on laatia toteutuskelpoinen tuotantosuunnitelma. Se ottaa huomioon markkinoiden tarpeet, tuotannon mahdollisuudet ja tukee yrityksen kokonaisstrategiaa. Tämä pitää käytännössä sisällään materiaalien, koneiden, ihmisten ja toimittajien ohjauksen. Erilaiset tuotantoympäristöt tarvitsevat erilaisia ERP-järjestelmiä. Tuotannonohjausjärjestelmä on lähes poikkeuksetta jokaisessa ERP-järjestelmässä.

Osto

Ostotoiminnot ovat ERP-järjestelmän ydinaluetta. Ostomoduuusta on toimivat rajapinnat tuotanto-, myynti- ja markkinointitoimintoihin. Toisissa ERP-järjestelmissä on saatavana integroitu asiakkuuden hallintajärjestelmä, jolloin se tukee erityisesti näitä toimintoja.

Myynti

Myyntitilastoja pystytään seuraamaan reaaliaikaisesti ERP-järjestelmän avulla. Raportit voidaan ottaa esimerkiksi tuotekohtaisesti, konekohtaisesti tai asiakaskohtaisesti; usein ne ovat saatavilla graafisina. Nykyisissä ERP-järjestelmissä myyntitilastotoiminnot toimivat hyvin.

Varastohallinta

ERP-järjestelmän varastohallinta voi sisältää tuotteiden, komponenttien ja muiden nimikkeiden varastopaikkojen ja varastosaldojen hallinnan, inventoinnit sekä varastojen sisäiset että niiden väliset tapahtumat. Yritykset vaativat myös usein reaaliaikaista inventointi mahdollisuutta.

Langaton ERP-järjestelmä

Langaton ERP-järjestelmä on käytössä joissakin yrityksissä; tämä perustuu mobiilipäätelaitteisiin. Langatonta ERP-järjestelmää käytetään liikkuvassa työssä ja se tuo seuraavia hyötyjä: asiakaspalvelun tehostuminen, tilaus-toimitus-laskutusketjun nopeutuminen ja työntekijän henkilökohtaisen tuottavuuden parantuminen.

[10]

4 ERP-JÄRJESTELMÄN ELINKAARI

4.1 ERP-järjestelmän elinkaaren vaiheet

ERP-järjestelmän elinkaari koostuu seuraavista vaiheista: päätösvaihe, hankintavaihe, toteutusvaihe, käyttö- ja ylläpitovaihe, kehitysvaihe ja lopetusvaihe.



Kuva 4. ERP-järjestelmän elinkaaren vaiheet ja näkökulmat.

ERP-järjestelmän eri näkökulmia ovat tuote, prosessi, ihmiset ja muutoksenhallinta. Kuva 4 havainnollistaa näitä asioita. Tässä kohdassa käydään nämä lyhyesti läpi.

[11]

Päätösvaihe

Tämän vaiheen aikana tehdään päätös hankkia ERP-järjestelmä. Yrityksen pitää kartoittaa millaisia tarpeita organisaatiolla on uudesta tietojärjestelmästä. Selvitetään millainen ERP-järjestelmä parhaiten tukee organisaation liiketoimintaa.

Hankintavaihe

Tässä vaiheessa valitaan markkinoilta sopiva ERP-järjestelmä, joka parhaiten vastaa organisaation vaatimuksiin. Näin minimoidaan ERP-järjestelmän räätälöinnin tarve.

Tässä voidaan käyttää konsulttiyritystä selvittämään vaatimuksia. Konsultointia tarvitaan seuraavissa vaiheissa, etenkin toteutusvaiheessa. Tässä yhteydessä tehdään ostosopimus valitun järjestelmätoimittajan kanssa; sopimuksessa on hyvä eritellä mahdolliset konsultointi, koulutus- ja ylläpitopalvelut. Tässä vaiheessa on myös tärkeää tehdä analyysi, mitä ERP-järjestelmän investointi tulee tuottamaan yritykselle.

Toteutusvaihe

Toteutusvaiheessa ERP-järjestelmä sovitetaan ja konfiguroidaan yrityksen toimintoihin. Yleensä tässä vaiheessa käytetään konsultteja, jotka tarjoavat palveluitaan toteutukseen. Yrityksen kannattaa myös miettiä järjestelmän konfiguroinnin koulutusta omalle henkilökunnalle, varsinkin jos yritys tulevaisuudessa suunnittelee liiketoimintaan muutoksia.

Käyttö- ja ylläpitovaihe

Tässä vaiheessa yritys saa hyötyä ERP-järjestelmän käytöstä ja yrityksen liiketoimintaprosessit on saatu toimimaan järjestelmän kanssa. Liiketoiminnan erityispiirteet sisällytetään järjestelmään mahdollisuuksien rajoissa. Mahdolliset ongelmatekijät ja erilaiset häiriöt on pyrittävä minimoimaan.

Kehitysvaihe

Tässä vaiheessa ERP-järjestelmä integroituu organisaatioon paremmin ja tarjoaa lisäarvoa yritykselle. Se tarjoaa uusia mahdollisuuksia liiketoimintaan, kuten toimitusketjun ja asiakkuuden hallintaa, työnkulkua ja aikataulutustoimintoja. Kehitysvaiheessa ERP-järjestelmä voidaan integroida yhteistyökumppanien tietojärjestelmiin.

Lopetusvaihe

Kun ERP-järjestelmä ei enää vastaa liiketoiminnan vaatimuksin ja tarjolla on uusia parempia järjestelmiä, tällöin yritys saattaa tehdä päätöksen, että nykyinen ERP-järjestelmä korvataan toisella tietojärjestelmällä, joka vastaa paremmin organisaation tarpeisiin.

[11]

4.2 ERP-järjestelmän elinkaaren näkökulmat

Seuraavassa määritellään neljä eri näkökulmaa, joiden eri vaiheissa ERP-järjestelmän elinkaarta tulisi analysoida. Ne ovat tuote, prosessi, ihmiset ja muutoksen hallinta. [11]

Tuote

Tämä näkökulma keskittyy erityisesti tuotteeseen liittyviin seikkoihin, laitteiston ja käyttöjärjestelmän vaatimuksiin, toiminnallisuuteen ja niihin liittyviin teknisiin yksityiskohtiin. Ohjelmiston ominaisuudet pitää ymmärtää hyvin, jotta voidaan selvittää, voidaanko ohjelmistoa käyttää tehokkaasti organisaatiossa ja kuinka sitä voidaan soveltaa organisaation tavoitteisiin.

Prosessi

Jokaisella organisaatiolla on oma ydiosaamisensa ja toiminnallisuutensa, joita ERP-järjestelmän on tuettava. ERP-järjestelmän pitää auttaa päätöksenteossa, jota tarvitaan kun hallitaan organisaation resursseja ja toimintoja. Yleensä tärkein asia ERP-järjestelmän investoinnissa ovat tuotannon ja suunnittelun prosessit, jotka organisaatio sopeuttaa uusiin liiketoimintamalleihin ja toiminnallisiin vaatimuksiin. Tällöin ERP-järjestelmä tarjoaa parhaan suorituskyvyn ja hyödyn yritykselle.

Ihmiset

Tämä näkökulma koskee henkilöresursseja, heidän roolejaan ja osaamistaan toiminnanohjausjärjestelmän elinkaareissa. Henkilöiden tietotaitoa ja rooleja pitää kehittää, jotta voidaan minimoida ERP-järjestelmän aiheuttama muutosvaikutus organisaatiossa. Epävarmuuden hallitseminen, käytäntöjen muuttuminen ja sopeutuminen uuteen toimintamalliin yrityksessä ovat asioita, jotka pitää hyväksyä.

Muutoksen hallinta

Tämä näkökulma koskee organisaation henkilöiden riittävää ammatillista osaamista muutoksen hallintaan. Tämä osaaminen ison tietojärjestelmän monimutkaisessa muutoksessa auttaa saamaan hyviä tuloksia, oikeaan aikaan ja oikein kustannuksin. Muutoksen hallinnan lähestymistapa pyrkii varmistamaan myös sen, että yrityksellä on valmiudet hyväksyä uusi ERP-järjestelmä ja riittävä ammatillinen valmius; tämä edesauttaa organisaatiota saamaan parhaan hyödyn järjestelmän käytöstä.

5 ERP-JÄRJESTELMÄN VAATIMUSMÄÄRITTELY PK-YRITYKSESSÄ

Vaatimusmäärittely luo pohjan toiminnanohjausjärjestelmä hankkeelle ja sen läpivienti onnistuneesti on tärkeää. Vaatimusten määrittelyyn on olemassa useita eri malleja ja menetelmiä, nämä eivät kuitenkaan suoraan sovellu loppukäyttäjille. Näiden mallien käyttö vaatii ammattitaitoa, joka ei välttämättä löydy yrityksestä. Pk-yrityksissä on haasteena rajalliset resurssit. Seuraavassa käydään läpi vaatimusmäärittelyn näkökulmia, sen sijoittautumista ERP-projektiin, sen vaiheita ja kehitystarpeita. [2]

5.1 Vaatimusmäärittely ERP-järjestelmän lähtökohtana

Vaatimusmäärittelyssä selvitetään, mitä uudelta ERP-järjestelmältä odotetaan, näitä ovat tavoitteet, tarpeet ja odotukset. Ne pyritään esittämään eri käyttäjien tai roolien mukaan luokiteltuna. Tarkoituksena on esittää se, mitä kehitettävältä järjestelmältä vaaditaan.

Vaatimusmäärittely on tärkeimpiä vaiheita koko uuden järjestelmän hankkeessa. Tämä vaihe määrittää, millainen lopullisesta järjestelmästä tulee. Seuraavat vaiheet rakentuvat tämän vaiheen pohjalle, joten puutteet ja virheet kostaantuvat seuraavissa vaiheissa. Tällöin näiden korjaaminen tulee huomattavasti kalliimmaksi. Vaatimustenmäärittely palvelee sekä loppukäyttäjää että toimittajaa.

Vaatimukset perustuvat yrityksen tavoitteisiin ja käyttäjien tarpeisiin. Tarve on ilmaisu siitä, mitä halutaan. Tässä vaiheessa tunnistetaan, kootaan, ryhmitellään, muokataan ja karsitaan tarpeet. Ne voidaan priorisoida teknisten ja taloudellisten tekijöiden perusteella. Vaatimukset luokitellaan välttämättömiin ja toivottaviin. Toteuttaminen on yleensä kompromissi näistä. Vaatimusten kustannukset selvitetään.

Vaatimukset jaetaan toiminnallisiin ja ei-toiminnallisiin. Toiminnalliset vaatimukset kuvaavat mitä toimintoja tai palveluita järjestelmältä vaaditaan. Ei-toiminnalliset vaatimukset liittyvät järjestelmän suorituskykyyn ja laatuominaisuuksiin.

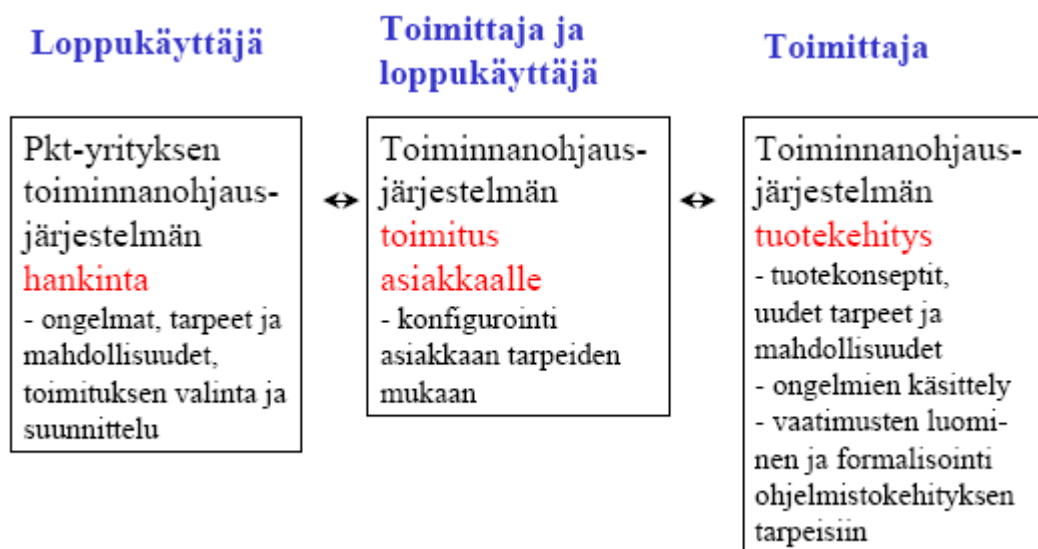
Jotkut vaatimukset ovat tavoitteita, joihin pyritään. Tällöin toteutustapa jätetään suunnittelijoille tai toimittajille. Toisissa vaatimuksissa voi olla suunnittelurajoitus. Rajoitus voi olla esimerkiksi tietty käyttöjärjestelmä tai lakivelvoite. Rajoitukset tulee miettiä tarkkaan, koska ne voivat olla este hyvälle lopputulokselle.

Vaatimusmäärittely tulee dokumentoida niin, että niiden toteutumista voidaan seurata koko hankkeen ajan. Tässä yhteydessä voidaan käyttää myös erilaisia mallipohjia, joista on helpompi lähteä liikkeelle.

[2]

5.2 Vaatimusmäärittelyn näkökulmat

Toiminnanohjausjärjestelmän hankkeessa voidaan tunnistaa kolme eri näkökulmaa vaativuusmäärittelyyn, kuten kuvassa 5 on esitetty.



Kuva 5. Toiminnanohjausjärjestelmän vaativuusmäärittelyn näkökulmat.

Toiminnanohjausjärjestelmän vaatimusmäärittely loppukäyttäjän näkökulmasta

Järjestelmän hanke voi perustua liiketoiminnan tarpeista lähtevään strategiaan, jossa on otettu huomioon tietotekniikan tuomat kustannus- ja aikasäästöt sekä uudet liiketoimintamahdollisuudet. Pk-yrityksissä hankkeen käynnistäjänä saattavat olla käytännön toiminnan ongelmat, ulkoiset muutokset asiakkailta, ympäristöstä tulevat vaatimukset tai halu seurata trendejä.

Toiminnanohjausjärjestelmän toteutus on asiakkaan ja toimittajan yhteistyöhanke

Vaatimukset selvitetään yhdessä toimittajan kanssa, siitä huolimatta, että asiakas olisi käynyt ne läpi jo ennen toimittajan valintaa. Toimittajan tuotteista, toimintatavoista ja asiakkaan valmiuksista riippuu, miten asiakkaan vaatimuksia käytetään ja kehitetään toteutusprojektissa.

Toiminnanohjausjärjestelmän tuotekehitys on toimittajan tekemää ohjelmistokehitystä

Ohjelmistokehitys ei ole irrallaan asiakasprojektista. Kehitysehdotukset tulevat usein asiakkaalta ja kehitystä saatetaan tehdä asiakasprojektin aikana. Vaatimusten esittämiseen ohjelmistokehityksessä on olemassa useita eri malleja ja menetelmiä. Ongelmana näissä on se, että loppukäyttäjät eivät välttämättä ymmärrä niitä.

[2]

5.3 Loppukäyttäjän näkökulma toiminnanohjausjärjestelmän vaatimuksiin

Yrityksen tietotekniikkaa voidaan kehittää kolmella eri tasolla:

- Strategian määrittelyllä;
- Yksittäisillä tietotekniikka hankkeilla;
- Järjestelmien ja toimintatapojen tehostamisella.

Tässä käsitellään vaatimusmäärittelyä yksittäisen tietojärjestelmähankkeen näkökulmasta ja lähtien tilanteesta, että yritys on tehnyt päätöksen hankkia uusi ERP-järjestelmä. Hankkeen tarpeiden ja vaatimusten määrittely jakaantuu kahteen seuraavaan vaiheeseen:

On tehty päätös hankkia uusi järjestelmä, mutta toimittajaa ja tuotetta ei ole vielä valittu

Tällöin tunnistetaan keskeisimmät tarpeet, käydään läpi toimintatapojen kehittäminen tietojärjestelmän hankintaan liittyen. Tässä ei kuitenkaan määritellä yksityiskohtaisesti miten järjestelmän tulisi toimia. Tämän vaiheen tulee antaa riittävästi tietoa, jotta toimittaja voidaan valita. Toiminnallinen kuvaus, jonka pohjalta voidaan pyytää tarjous toimittajalta ja selvittää tarjolla olevien tuotteiden soveltuvuus on oleellinen. Tarpeiden kartoitus on hyvä tehdä tässä vaiheessa ja vasta tämän jälkeen valitaan toimittaja. Jos näin ei toimita, saattaa tulla ongelmia ja lisätyötä hankintaprojektin aikana ja käyttövaiheessa.

Toimittajan ja ohjelmistotuotteen valinnan jälkeen tehdään tarkempi määrittely uudesta järjestelmästä

Tässä selvitetään, miten valittu tuote sovitetaan yrityksen toimintaan. Tässä yhteydessä tunnistetaan käyttäjäroolit, käyttäjien tehtävät, prosessit ja keskeiset henkilöt. Tässä voidaan käydä läpi vielä vaatimuksia, mutta varsinainen vaativuusmäärittely pitäisi olla jo tehty ennen tuotteen hankintaa.

[2]

5.4 Järjestelmätyypit ja niiden vaikutus vaatimusmäärittelyyn

ERP-järjestelmät voidaan jakaa kolmeen seuraavaan luokkaan:

Räätälöity yrityksen tarpeisiin

Järjestelmä kehitetään kokonaan asiakkaiden tarpeiden mukaan. Tässä mallissa vaatimusmäärittelyn rooli korostuu, koska ei lähdetä liikkeelle valmiista ratkaisusta. Ratkaisun hyvänä puolena on, että saadaan juuri sellainen järjestelmä kuin halutaan. Ongelmana on kehittämisen ja ylläpidon vaatimat suuret resurssit asiakkaalta ja toimittajalta. Hankkeen viivästymisen ja epäonnistumisen vaara on myös olemassa. Räätälöidyt ratkaisut tulevat harvoin kysymykseen pk-yrityksissä. Tällainen ratkaisu on hyvin kallis.

Esikonfiguroidut ja parametroitavat järjestelmät

ERP-järjestelmän yleisin menettelytapa on käyttää standardituotteita, joista järjestelmä luodaan konfiguroimalla. Tämä tarkoittaa moduulien valintaa ja järjestelmän sovittamista asiakkaan tarpeisiin parametrionnin avulla. Parametreilla valitaan esimerkiksi asiakkaalle sopiva toimintatapa (autoteollisuus, vähittäiskauppa ja niin edelleen) ja niillä voidaan asettaa laskentatapoja, raportointitapoja tai muokata käyttöliittymää.

Täysin standardit tuotteet

Tässä mallissa toimitetaan aina täsmälleen sama järjestelmä. Myös nämä tuotteet vaativat yrityksen perusdatan syöttämisen, joten raja parametroitavan ja standardituotteen välillä on hämärä. Tämä malli sopii tarkasti rajattuihin toimialoihin.

Muuta huomioitavaa

Järjestelmätyypin jaon tekee vielä vaikeaksi se, että toimitettava järjestelmä joudutaan usein liittämään toisiin olemassa oleviin järjestelmiin, jotka usein vaativat räätälöintejä.

Jos yrityksellä on ennestään hyvät suhteet toimittajaan, niin toimittaja voi olla hankintaprosessissa mukana jo varsin aikaisessa vaiheessa. Tämä voi tulla kysymykseen, kun yrityksessä on jo käytössä ERP-järjestelmä, jota halutaan laajentaa tai päivittää. Huonona puolena tässä on, että vaatimusmäärittely ei saa riittävää huomiota ja ongelmiin haetaan ratkaisua toimittajan tuotteista. Tämä ei kuitenkaan ole välttämättä huono ratkaisu, vaikka pidemmän tähtäimen suunnitelmat jäävät helposti vähemmälle huomiolle.

Jos yrityksen käytössä olevat järjestelmät eivät voi toimia pohjana tai ei ole ennestään riittävän kattavaa järjestelmää, tällöin on yrityksen käynnistettävä hankintaprojekti, johon kuuluu tarpeiden määrittely. Tämä on usein vaativa tehtävä pk-yritykselle ja yritykset voivat joutua ”toimittajan vietäväksi”. Hyvän kuvan tuotteista yritys saa kokeilemalla ja testaamalla ohjelmistoa omalla aineistollaan, jos se on mahdollista. Myös tutustuminen muiden yritysten vastaaviin järjestelmiin ja käyttökokemuksiin on hyödyksi. Toimittajan referenssien ja laatu järjestelmän arviointi saattaa tulla kysymykseen.

Suurilla ERP-järjestelmien toimittajilla olevat ratkaisut voivat olla pitkälle mietittyjä. Konfiguroitavista ratkaisusta voi olla olemassa valmiit prosessimallikirjastot, joista yritykselle valitaan sopivin toimintatapa. Prosessimallikirjastojen kasvaessa ja monimutkaistuessa pk-yrityksille tämän vaihtoehdon valinta voi olla haastavaa. Käytettävät välineet on usein kuitenkin suunniteltu tietotekniikkaa hyvin ymmärtävien osapuolien tueksi.

ERP-järjestelmän hankintaa tulisi tarkastella kokonaisuutena, eikä pelkästään teknisenä muutoksena. Vaatimusmäärittelyssä tulee lähteä liikkeelle liiketoiminnan ja käyttäjien tarpeesta.

ERP-järjestelmien hankkeiden laajuus ja vaikuttavuus vaihtelee organisaatiossa laajoista uudelleenjärjestelyistä puhtaisiin tietojärjestelmäostoihin. Ei ole olemassa sääntöjä kuinka laajasti ja syvällisesti vaatimusmäärittely tulee tehdä.

Seuraavat ääripäät voidaan tunnistaa:

- Liian laaja tarpeiden kartoitus. Tämä tulee kalliiksi ja sen läpikäynti vie paljon aikaa. Liian laaja tarpeiden kartoitus myös viivästyttää hanketta.
- Jos hankitaan uusi järjestelmä pohtimatta lainkaan, mihin sitä tarvitaan ja miten sen kanssa toimitaan, käyttöönotto viivästyy ja kuluttaa paljon resursseja. Pahimmassa tapauksessa järjestelmää ei saada lainkaan käyttöön tai se jää vajaakäyttöön.

ERP-järjestelmän implementointiin vaikuttavat hankkeen luonne ja tavoitteet, organisaation koko, osaaminen, yrityskulttuuri ja käytettävissä olevat resurssit. Pk-yrityksissä resurssit ovat rajallisia, kun taas osaamistaso ja yrityskulttuuri saattavat vaihdella.

[2]

5.5 Vaatimusmäärittelyn vaiheet

Tavoitteiden määrittely

ERP-järjestelmän implementointi on suuri ja pitkävaikutteinen investointi, joten on tärkeää käydä läpi kehityssuuntia ja pitkän tähtäimen tavoitteita. Tämä varmistaa, että järjestelmä ei estä jatkossa yrityksen kehittymistä.

Tavoitteiden saavuttamista auttaa, jos voidaan määritellä konkreettiset tavoitteet. Näitä ovat esimerkiksi prosessien ja työnkulkujen nopeuttaminen, päällekkäisen työn vähentyminen, toimipisteiden yhteistyön paraneminen, tietojen syöttö vain yhteen kertaan ja tietojen parempi hyödyntäminen. Tavoitteiden tarkastelu vaatimusmäärittelyssä ja implementoinnin aikana auttaa yritystä keskittymään niihin asioihin, jotka ovat oleellisia.

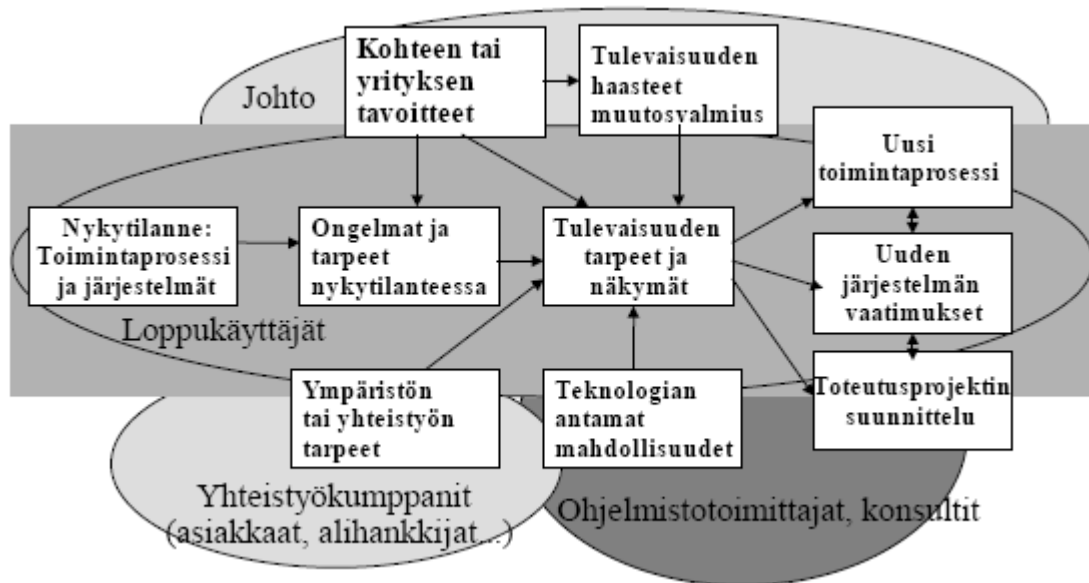
Tarpeiden tunnistus

Yrityksellä on yleensä, jokin käsitys siitä, mihin järjestelmää tarvitaan. Nämä tarpeet eivät ole kuitenkaan usein ole niin selkeitä, jotta ne voitaisiin kerätä vaatimuksiksi. Tarpeet pitää yleensä tunnistaa, poimia, perata ja kypsyttää, jotta niistä tulee vaatimuksia. Se on eräänlainen prosessi. Kuvassa 6 on esitetty, mitä pk-yrityksen tulisi käydä läpi ERP-järjestelmähankkeen tarpeita selvittäessä.

On tärkeää tiedostaa, mistä yrityksen kilpailukyky muodostuu ja mitkä ovat liiketoiminnan tavoitteet. Tämä mahdollistaa sen, että järjestelmän avulla pystytään seuraamaan ja ohjaamaan näitä tekijöitä. Nykyiset toimintaprosessit ja ongelmat tulee tunnistaa. Olemassa olevat järjestelmät tulee ottaa myös huomioon.

Tulevaisuuden vaatimukset tulee ottaa huomioon. Muutoksia saattaa tulla organisaatioon, tuotteisiin ja tuotantolaitteisiin. ERP-järjestelmän tulisi olla joustava, jotta sitä pystytään muokkaamaan yrityksen vaatimuksien tai toimintaympäristön muuttuessa. ERP-järjestelmän muokkaaminen olisi hyvä pystyä tekemään yrityksen omin resurssein.

Kun ERP-järjestelmän tarpeita kartoitetaan, tulee miettiä, miten järjestelmä tukee yhteistyötä muihin yrityksiin, asiakkaisiin, alihankkijoihin tai toimittajiin. Samoin pitää selvittää, mitä ja miten informaatiota välitetään ja käsitellään.



Kuva 6. Vaatimusten tunnistaminen pk-yrityksen ERP-järjestelmän hankinnassa.

Tarpeiden tunnistus luo pohjan uusien toimintaprosessien määrittelylle ja vaatimuksille. Kuvassa 6 nähdään myös, mitkä osapuolet ottavat osaa mihinkin pohdintavaiheeseen.

Tarpeista vaatimuksiksi

Tarpeiden kartoitusta tehdään yleensä useilla menetelmillä ja monissa ryhmissä. Tästä johtuen ne voivat sisältää päällekkäisiä ja ristiriitaisia asioita. Vaatimusten käsittely on iteratiivinen prosessi, jota voidaan tehdä listaamalla, ryhmittelemällä ja priorisoimalla tarpeita. Aikaisemmat konkreettiset tavoitteet auttavat priorisoinnissa.

Tässä vaiheessa on tärkeää tiedostaa toimintakokonaisuus, koska tarpeet muodostuvat eri toiminnoista. Samoin pitää selvittää miten eri tarpeet yhdistetään, miten kokonaisprosessi toimii ja mikä on järjestelmän rooli. Tässä voidaan käyttää mallintamismenetelmiä, joka vaatii näitä tukevien välineiden hallintaa. Tässä yhteydessä voidaan käyttää ohjelmistotoimittajan tai konsultin apua prosessien kuvaamiseen.

Tarpeiden läpikäynnissä tulee selvittää, mikä on ihmisen ja järjestelmän rooli kussakin toiminnassa. Näin saadaan selville automatisoinnin tarve. On huomattavaa kuitenkin, että monissa toiminnoissa tarvitaan ihmisen päätöksentekoa ja osallistumista.

Vaatimusten dokumentointi

Analysoinnin, muokkauksen ja priorisoinnin tuloksena saatavat tavoitteet dokumentoidaan. Dokumentointia voidaan käyttää oman organisaation tukena ja kun ollaan yhteydessä ohjelmistotoimittajiin. Keskeiset vaatimukset voidaan liittää tarjouspyyntöön ja pyytää toimittajaa ottamaan kantaa siihen, miten heidän tuotteensa täyttää nämä vaatimukset. Dokumentaatioon liitetään mahdolliset prosessikuvaukset ja prosessimallit.

[2]

5.6 Yhteenveto vaatimusmäärittelystä

Valmiudet vaatimusten määrittelyyn vaihtelevat pk-yrityksissä. Ongelmana on usein, että tarpeet ovat irrallisia toisistaan ja pk-yrityksessä ei ole pystytty hahmottamaan kokonaisprosesseja, miten yritys kokonaisuutena toimii ja mikä vaikuttaa mihinkin. Jos prosessia ei tunneta, silloin ei myöskään ymmärretä toiminnanohjausjärjestelmän eri osien ja toimintojen merkitystä, joten järjestelmän hyödyt voivat jäädä saavuttamatta. Kun ohjelmistotoimittaja pystyy esittämään mahdollisimman konkreettisesti, miten järjestelmä tukee yrityksen toimintaa ja millaisia toimintatapoja se mahdollistaa tai vaatii, tällöin siitä on hyötyä yritykselle. Tässä on mahdollisuutena kuvata tuotetta ja sen eri vaihtoehtoja prosessi-, työnkulku- tai tietovuomalleilla.

Mallien hyväksikäyttö ei ole kuitenkaan itsetarkoitus. Mallit lisäävät ymmärrystä siitä mitä ollaan tavoittelemassa ja miten saadaan mahdollisimman hyvä toteutus. Malleja sovellettaessa pk-yrityksessä tulee löytää oleelliset asiat ja sopiva esitystaso. Samojen prosessimallien käyttö läpi koko ERP-järjestelmän implementoinnin totuttaisi organisaation toimijat prosessiajatteluun ja loisi pohjaa yhteiselle näkemykselle.

Vaatimusmäärittely luo pohjan järjestelmän tekniselle ja organisatoriselle käyttöönotolle. Pk-yrityksen hankkiessa toiminnanohjausjärjestelmän usein sen realistinen mahdollisuus on ostaa valmis pakettiratkaisu, joka enemmän tai vähemmän vastaa sen liiketoimintaprosesseja. Tämän takia tarpeita ja vaatimuksia ei kannata hioa liian pitkälle ennen yhteydenottoa ERP-järjestelmien toimittajiin. Vaatimusten esittämiseen tulisi löytää sopivat esitystavat ja menetelmät, joiden pohjalta voidaan keskustella toimittajan kanssa tarjousvaiheessa. Tarjousvaiheessa voidaan vielä tarkentaa vaatimuksia. Tärkeää on, että yritys saa konkreettisesti selville miten ERP-järjestelmä tukee yrityksen kokonaisprosesseja. Tarpeen on myös selvittää mitä ERP-järjestelmän implementointi vaatii yritykseltä ja miten järjestelmä tulee muuttamaan yrityksen toimintatapoja.

[2]

6 ERP-PROJEKTIN ONNISTUMISEN EDELLYTYKSET

Yritykset joutuvat nykyisin kohtaamaan karun todellisuuden: ennakoida, vastata ja reagoida markkinoiden kasvaviin vaatimuksiin. Tiukassa kilpailutilanteessa, liiketoimintastrategia ei vain ratkaise onnistumista, vaan se koskee myös yrityksen olemassaoloa. Nyt, enemmän kuin koskaan, tehokas liiketoimintastrategia keskittyy aggressiivisesti ja tehokkaasti hyödyntämään tietotekniikan käyttöä.

ERP-järjestelmä tukee ja kehittää yrityksen prosesseja ja vähentää kustannuksia. ERP-järjestelmään saadaan muun muassa linkitettyä verkkoliiketoiminta ja toimitusketjun hallinta. Yhdistämällä toimitusketjut muiden yritysten järjestelmiin läpimenoaika saadaan lyhemmäksi ja varastot pidettynä pieninä.

Asioilla on kuitenkin aina kaksi puolta. Todellisuudessa ERP-järjestelmä on kuitenkin kallis ja iso investointi yritykselle. Useasti pitää käyttää konsultointia ERP-järjestelmän hankkeen toteutuksessa. ERP-järjestelmä on monimutkainen, moduulit ovat toisiinsa sidotut ja järjestelmän prosessit pitää sovittaa organisaatioon. Yleensä yrityksillä on heidän omat kilpailuetujaan tuottavat prosessit. Yritykset joutuvat muuttamaan heidän hyväksi koettujaan prosesseja sopivaksi toiminnanohjausjärjestelmään. Näin ne saavat parhaan hyödyn ERP-järjestelmästä ja voivat hyödyntää ERP-järjestelmän ohjelmistoa myös tulevaisuudessa. Seuraavat asiat ovat keskeisiä onnistuneen ERP-järjestelmän implementoinnissa.

Tiimityö ja tiimin koostumus

Tiimin yhteistyö on tärkeää koko ERP-järjestelmäprojektin ajan. Tiimin kokoaminen henkilöistä, joilla on erilainen toiminnallinen tausta, on haasteellinen. Tiimin pitäisi olla sekoitus omaa henkilökuntaa ja ulkopuolisia toimijoita. Oman organisaatioon kuuluvilla pitää olla teknistä osaamista suunnittelusta ja projektikokemusta. Sekä liiketoiminnan ymmärtäminen että tekninen osaaminen ovat välttämättömiä onnistuneen lopputuloksen saavuttamisessa.

ERP-järjestelmäprojektilla pitäisi olla korkea prioriteetti ja työkuorma suhteutettu tiimin henkilöstöön. Projekti henkilöiden pitäisi olla kokopäivätyössä ja sijainti lähellä toisia yhteydenpidon helpottamiseksi. Tiimille pitää järjestää omat toimitilat, jos se on mahdollista.

Tiimi pitää palkita ja sen pitää saada työstään kannustusta, jos he pysyvät aikataulussa ja budjetissa. Tiimin olisi tunnettava yrityksen toiminnot ja tuotteet, jotta he pystyvät tukemaan tärkeitä liiketoimintaprosesseja ERP-järjestelmällä.

Tiedon jakaminen yrityksen sisällä, erityisesti projektitiimin välillä sekä yhteistyökumppaneiden kesken on tärkeää. Tämä antaa edellytykset luottamukselle ja yhteistyöhengelle. Säännöllisiä palaveria on pidettävä tiimin kesken. Riskien ja vastuun jakaminen helpottavat tavoitteen saavuttamisessa.

Johdon tuki

Ylimmän johdon tukea tarvitaan läpi koko ERP-järjestelmäprojektin ajan. Hankkeen on saatava ylimmän johdon hyväksyntä. Strategisten liiketoiminnan tavoitteita on yhdenmukaistettava tukemaan ERP-järjestelmäprojektia. Nämä voidaan saavuttaa sitomalla johdon bonukset projektin onnistumiseen.

Ylimmän johdon pitää julkisesti ja yksiselitteisesti tunnustaa hankkeen tärkeys. Johtohenkilöt omalla osallistumisellaan osoittavat projektin tärkeyden. Johdon pitää taata, että projekti saa käyttöönsä tarvittavat henkilöresurssit ja riittävästi aikaa toteutukseen.

Esimiesten tulee tukea uusia päämääriä ja tavoitteita. Uudet organisaatorakenteet, roolit ja vastuut on vahvistettava. Johdon tulee hyväksyä uudet käytännöt ja menettelytavat. Organisaation ja uuden järjestelmän aiheuttamat muutokset tulee tiedottaa työntekijöille. Johdon tulee selvittää mahdolliset erimielisyydet projektissa.

Liiketoimintasuunnitelma ja visiot

Selkeä visio ja liiketoimintasuunnitelma on tarpeellinen ohjaamaan yrityksen suuntaa koko ERP-järjestelmäprojektin ajan. Liiketoimintasuunnitelma, jossa esitellään ehdotetut strategiset ja konkreettiset edut, resurssit, kustannukset, riskit ja aikataulu on tärkeä. Tämä auttaa organisaatiota keskittymään liiketoiminnan hyötyihin.

Pitää olla selkeä liiketoimintamalli, miten organisaatiota ohjataan projektin aikana. Pitää varmistaa, että yritystä ohjataan oikeaan suuntaan. Muutosten hallintaan pitää varata resursseja ja ongelmat pitää ratkaista. ERP-järjestelmäprojektin tavoitteet ja hyödyt tulee tunnistaa. Liiketoimintasuunnitelman tulee vaikuttaa työn tuottavuuteen positiivisesti ja tehdä työstä helpompaa.

Viestintä

Tehokas viestintä on välttämätöntä ERP-järjestelmäprojektissa organisaation kaikilla tasoilla. Viestinnän hallinta on tärkeää koko organisaatiossa. Projektitiimin tulee säännöllisesti tiedottaa projektin eteneminen muulle organisaatiolle. Keskijohdon kommunikoimilla on suuri merkitys, koska he toimivat ylimmän johdon ja työntekijöiden välimaastossa. Työntekijöille pitää etukäteen selvittää projektin laajuus, tavoitteet ja ennako-odotukset. Käyttäjien palaute, vaatimukset, kommentit ja reaktiot tulee ottaa huomioon.

Projektin hallinta

Hyvä projektinhallinta on olennaista. Tämä sisältää järjestelmien toteutuksen, osallistumisen liiketoimintayksiköiden ja liiketoimintaprosessien uudelleensuunnitteluun. Ehdotetut muutokset olisi arvioitava liiketoiminnallisia hyötyjä vasten, niin pitkälle kuin mahdollista ja toteuttaa myöhemmässä vaiheessa. Lisäksi ehdotetut tavoitteen muutokset on arvioitava suhteessa lisäaikaan ja kustannuksiin.

ERP-järjestelmäprojekti pitää virallistaa ja määritellä sen välitavoitteet. Hankkeen kriittiset polut ja ongelmakohdat tulee selvittää ja määritellä. Määräajoissa pysyminen helpottaa aikataulun ja budjetin rajoissa pysymistä ja projekti säilyttää uskottavuutensa. Projektin aikataulun päivittäminen tulee hoitaa. Jos aikataulusta ollaan myöhässä, tulee tehdä uusi realistinen aikataulu ja tiedottaa se läpi koko organisaation.

Projektinhallinnan on hyvä koordinoita koulutusta ja valvoa työntekijöiden osallistumista niihin. Määritellyt tehtävät pitää suunnitella hyvin. Tuloksien, aikataulun seuranta ja niiden vertaaminen budjetin tavoitteisiin on myös tärkeää. Projektin ongelmatilanteet tulee hallita ja ottaa huomioon. Onnistuneet tulokset on tärkeää tiedottaa aikaisessa vaiheessa.

Projektin johtajat

Projektille on hyvä valita projektijohtaja, joka asettaa tavoitteet ja hyväksyy muutokset. Liiketoimintajohtajan tulee olla vastuussa liiketoiminnan näkökulmasta. Uudistusta tukevat johtajat ovat tärkeitä projektin onnistumisen kannalta. Johtajien on pyrittävä ratkaisemaan erimielisyyksiä ja hoitamaan projektin saamaa vastustusta.

Projektijohtajan sitoutuneisuus on elintärkeää. Hänen pitää johtaa ERP-järjestelmäprojektia koko sen elinkaaren. Projektijohtajalla tulee olla ylin päätäntävalta koko ERP-järjestelmän projektissa läpi organisaation.

Soveltuva liiketoiminta ja aikaisemmat liiketoimintajärjestelmät

Soveltuva liiketoiminta ja aikaisempien liiketoimintajärjestelmien huomioon ottaminen on tärkeää ERP-järjestelmän elinkaaren alusta alkaen. Vakaa ja menestyksellinen liiketoiminnan toimivuus ovat oleellisia ERP-järjestelmässä. Liiketoimintaan ja informaatiojärjestelmiin liittyvät olemassa olevat liiketoimintaprosessit, organisaatorakenne, yrityskulttuuri ja tietotekniikka vaikuttaa onnistumiseen.

Muutoksen hallinta ja yrityskulttuuri

Muutoksen hallinta on tärkeää, alkaen hankintavaiheesta ja jatkuen läpi koko ERP-järjestelmän elinkaaren. Yrityskulttuurin ja organisaation muutokset ovat tärkeitä hallita. Yrityskulttuuri, johon kuuluvat organisaation yhteiset tavoitteet ja arvot ovat tärkeitä. Organisaation pitäisi olla avoin muutoksille. Laadun painotus, hyvä arviointikyky ja vahva halu hyväksyä uusi teknologia edistävät ERP-järjestelmän implementointia. Johdon tulee olla sitoutunut käyttämään ERP-järjestelmää; tämä auttaa liiketoiminnan tavoitteiden saavuttamisessa. Käyttäjien erilaiset tarpeet pitää ottaa huomioon ja ongelmat on käytävä läpi säännöllisissä palaverissa.

Osana muutoksenhallintaa käyttäjien tulee olla mukana suunnittelemassa ja toteuttamassa ERP-järjestelmän liiketoimintaprosesseja. Tämä mahdollistetaan sopivalla koulutuksella. Koulutuksen pitäisi olla ensisijainen ERP-järjestelmänprojektin alusta alkaen. Tarvittava aika ja resurssit pitää järjestää erilaisiin ERP-järjestelmän koulutuksiin. Tietotekniikan asiantuntijoiden koulutus ja ammatillinen kehitys ovat keskeisessä roolissa. Investointeja koulutukseen, suunnitteluun ja menetelmien kehitykseen pitää tehdä. Työntekijät tarvitsevat myös koulutusta ymmärtää, miten järjestelmä tulee muuttamaan liiketoimintaprosesseja. Koko organisaation tulee tukea ja kannustaa koulutukseen positiivisessa hengessä.

Liiketoimintaprosessien muokkaaminen ja ohjelmistomuutoksien minimointi

ERP-järjestelmän muutoksien minimointi ja yrityksen liiketoimintaprosessien uudelleensuunnittelu ovat tärkeitä ERP-järjestelmän elinkaaren alusta alkaen. On väistämätöntä, että yrityksen liiketoimintaprosessit sulautetaan uuteen järjestelmään. Organisaation tulisi olla valmis muuttamaan liiketoimintaansa, siten että ohjelmistomuutoksia tulee mahdollisimman vähän. Jos ERP-järjestelmää joudutaan räätälöimään niin ohjelmistovirheiden riski kasvaa. ERP-järjestelmän muutokset hankaloittavat myös uusien versioiden ja päivityksien saatavuutta. Prosessien mallintaminen työkalujen avulla helpottaa liiketoimintaprosessien muokkaamista, ilman että joudutaan muuttamaan ohjelmakoodia.

Laaja liiketoimintaprosessien uudelleensuunnittelu pitäisi aloittaa jo ennen uuden ERP-järjestelmän valintaa. Kun konfiguraatiota ollaan tekemässä, pitää panostaa uudelleensuunnitteluun iteratiivisesti hyödyntäen uuden järjestelmän parannuksia. Sitten, kun järjestelmä on käytössä, uudelleenmuokkausta tulee jatkaa ottaen huomioon uudet parannusehdotukset ohjelmaan. ERP-järjestelmätoimittajan kokemusta aikaisemmista toimituksista kannattaa myös hyödyntää.

Ohjelmistokehitys, testaus ja ongelmanratkaisu

Ohjelmistokehitys ja ohjelmistotestaus pitää aloittaa ERP-järjestelmän toteutusvaiheessa. ERP-järjestelmän arkkitehtuuri tulee määrittää ennen käyttöönottoa ottaen huomioon tärkeimmät vaatimukset toteutuksessa. Tästä on se hyöty, ettei ERP-järjestelmän uudelleenkonfigurointia tarvitse suorittaa jokaisessa implementoinnin vaiheessa.

Jotta ERP-järjestelmästä saadaan mahdollisimman suuri hyöty, se tulee integroida olemassa oleviin ja mahdollisiin uusiin ohjelmistoihin. Liitännät toisiin järjestelmiin voidaan joutua itse kehittämään, jos ne eivät ole yleisesti saatavilla tai jo valmiina ERP-järjestelmässä.

Ohjelmistotestaus on tärkeä osa-alue, johon pitää panostaa. ERP-järjestelmän ohjelmisto-ongelmia selvittäessä kannattaa hyödyntää toimittajaa ja konsultteja. Kärsivällisyys, sitkeys ja ongelmanratkaisu ovat oleellisia asioita ohjelmointitestauksessa.

Seuranta ja suorituksen arviointi

Lopuksi, seuranta ja suorituksen arviointi ovat tärkeässä roolissa ERP-järjestelmän käyttö- ja ylläpitovaiheessa. Projektin edistymistä on tärkeää seurata aktiivisesti. Pitää asettaa väli- ja lopulliset tavoitteet. Nämä pitäisi mitata vasten projektin tavoitteita. Tavoitteet tulee olla mahdollista saavuttaa ja onnistumiset tulee palkita.

Seuraavia kahta mittaus kriteeriä voidaan käyttää. Projektinhallinnan mitattavat kriteerit tulisi perustua aikataulun toteutukseen, kustannuksiin ja laatuun. Toiminnallisia kriteereitä olisi käytettävä mittaamaan tuotantoa vasten. Seuranta ja palaute perustuu projektiryhmän ja käyttäjien analyysiin uudesta ERP-järjestelmästä.

Yhteenveto ERP-projektin onnistumisen edellytyksistä

Tärkeät ERP-projektin menestystekijät on käyty läpi. Onnistuneeseen lopputulokseen vaikuttavat oikea tiimin koostumus, sisältäen yrityksen omat työntekijät, toimittajan työntekijät ja konsultit. Hyvä kommunikointi näiden henkilöiden ja ryhmien kesken on

olennaista. Koska ERP-järjestelmään kuuluu monenlaisia toiminnallisia alueita, on myös tärkeää luoda vastaavat toiminnalliset rajat tiimin. Yhteistyö ja luottamus näiden ryhmien ja tiimin jäsenten välillä on elintärkeää. Myös muutoksenhallinta ja yrityskulttuuri ovat tärkeitä. Yrityskulttuuri, jossa työntekijät jakavat yhteiset arvot, tavoitteet ja ovat valmiit muuttamaan toimintatapojaan, ovat merkittäviä projektin kannalta. Lisäksi, käyttäjien koulutusta ja tukea pitäisi olla saatavilla ja käyttäjiä pitää kannustaa koulutukseen.

[12]

7 AVOIMEN LÄHDEKOODIN OHJELMISTO

Avoimen lähdekoodin ohjelmisto lyhenne on suomenos englannin kielen sanoista Open Source Software (OSS). Avoimen lähdekoodin periaate tarkoittaa yleisesti, että ohjelman lähdekoodi on vapaasti käytettävissä.

Avoimen lähdekoodin ohjelmat kilpailevat suljettujen, kaupallisten ohjelmien kanssa ja ne voivat tarjota edullisen ratkaisun käyttäjille, joilla on rajoitettu budjetti. Avoimen lähdekoodin ohjelmien jakelu on yleisesti ottaen vapaata, ainoastaan palvelut ja lisäominaisuudet ovat maksullisia, esimerkiksi asennus, räätälöinti, koulutus ja tekninen tuki. Kun uuteen tietokoneeseen asennetaan käyttöjärjestelmä tai joku muu ohjelmisto, pitää ostaa lisenssi. Avoimen lähdekoodin ohjelmisto voidaan asentaa omalle koneelle ilman kaupallista lisenssiä. Jos yritys menee konkurssiin tai lopettaa tuen tuotteelle, käyttäjillä ei ole enää turvaa. Ohjelmistovirheiden korjaukset, joita on yleensä paljon, on täysin riippuvainen organisaatiosta, joka myy tuotetta. Avoimen lähdekoodin ohjelmistojen etuna on nopea kehitys. Avoimen lähdekoodin ohjelmiston käyttäjät voivat saada tukea muilta käyttäjiltä ja he voivat kehittää ohjelmaa ja korjata mahdollisia virheitä. Avoimen lähdekoodin ohjelmistokehitykseen on vaikuttanut positiivisesti se, että ohjelmoijat kehittävät ohjelmia omiin tarpeisiinsa, jolloin heidän motivaationsa on korkea.

Avoimen lähdekoodin ohjelmistot ovat alkaneet syrjäyttää suljettuja, kaupallisia ohjelmistoja; esimerkiksi Linux-käyttöjärjestelmä on www-palvelin käytössä hyvin suosittu. Myös monet muut avoimen lähdekoodin ohjelmat ovat tulleet tunnetuiksi, esimerkiksi Firefox-selain, VLC-mediasoitin ja GIMP-kuvankäsittelyohjelma. Myös liiketoiminnassa on käytössä useita avoimen lähdekoodin ratkaisuja, kuten Apache-palvelin, MySql-relaatiotietokantaohjelma ja OpenJDK-Java-ohjelmointialusta. Luvussa yhdeksän esitellään muutamia yleisempiä avoimen lähdekoodin ERP-ohjelmistoja.

[13; 14; 15]

7.1 Avoimen lähdekoodin määritelmä

Tässä käydään läpi avoimen lähdekoodin määritelmä (Open Source Definition), virallinen englanninkielinen versio on löydettävissä muun muassa Open Source Initiative:n (OSI) internet-sivuilta <http://opensource.org/osd>. Open Source Initiative-järjestön tarkoitus on edistää avoimen lähdekoodin ohjelmien käyttöä ja kehitystä. Avoimen lähdekoodin määritelmä ei ole ohjelmiston lisenssi itsessään. Se on erittely siitä, mitkä ovat vaatimukset ohjelmistoille, joita kutsutaan avoimen lähdekoodin ohjelmistoiksi. Avoin lähdekoodi ei tarkoita pelkästään, että lähdekoodi on vapaasti saatavilla. Avoimen lähdekoodin ohjelmiston on täytettävä kaikki seuraavat vaatimukset:

- Vapaa levitysoikeus. Lisenssi ei saa rajoittaa kenenkään osapuolen myymistä tai jakeluoikeutta. Ohjelman täytyy olla vapaasti levitettävissä ja välitettävissä;
- Lähdekoodi. Ohjelman pitää sisältää myös sen lähdekoodi ja jakelu pitää sallia sekä lähdekoodina että käännetyssä muodossa. Lähdekoodi on oltava sellaisessa muodossa, että ohjelmoitsijan on mahdollista muokata sitä. Lähdekoodi ei saa olla tahallisesti epäselvä;
- Johdannaiset teokset. Lisenssin on sallittava muutokset ohjelmaan ja niistä johdannaiset teokset. Lisenssin täytyy sallia, että johdannainen ohjelma on levitettävissä samoin ehdoin kuin lisenssin alkuperäinen ohjelmisto;
- Lähdekoodin yhteenkuuluvuus. Lisenssi voi rajoittaa lähdekoodin levittämistä muunnettuna vain, jos lisenssi sallii korjaustiedostojen ja niiden lähdekoodin levittämisen. Korjaustiedostojen tarkoituksena on ohjelman muuttaminen käännösvaiheessa. Lisenssin on sallittava muutetusta lähdekoodista käännettyjen ohjelmien levittäminen. Lisenssi voi vaatia, että johdannaiset teokset pitää tehdä toisella nimellä tai niissä on käytettävä versionumerointia;
- Lisenssi ei saa syrjiä mitään henkilöä tai henkilöryhmää;
- Lisenssi ei saa syrjiä mitään liiketoiminta-alaa tai tutkimusta. Esimerkiksi, jos sitä halutaan käyttää geenitutkimukseen, se pitää sallia;
- Lisenssin jakelu. Oikeudet ohjelman lisenssiin koskee kaikkia, joilla ohjelma on käytössä ja erillistä lisenssiä ei tarvita;

- Lisenssi ei saa olla tuotekohtainen. Ohjelmaan liitettyjen oikeuksien ei tule olla riippuvaisia siitä, että ohjelma on osa jotakin tiettyä ohjelmistoa. Jos ohjelma poistetaan ohjelmistosta ja sen jälkeen käytetään tai levitetään ohjelman lisenssillä, kaikkien osapuolten, joille ohjelmaa levitetään, tulee saada alkuperäisen ohjelmistopakettin oikeudet;
- Lisenssi ei saa asettaa rajoituksia muille ohjelmille, jotka levitetään yhdessä lisensoidun ohjelmiston kanssa.

Avoimen lähdekoodin ohjelmistossa tekijänoikeudet ovat koodin kirjoittajalla, mutta ohjelma on lisensoitu kaikille avoimen lähdekoodin lisenssillä. Tämän seurauksena kenellä tahansa on oikeus levittää lähdekoodia, käyttää sitä hyväksi ohjelmiston kehittämisessä ja levittää muokattua ohjelmaa ja lähdekoodia eteenpäin.

[15; 16; 17]

7.2 Ohjelmistolisenssit

Ohjelmistoliiketoiminnassa on käytössä useita erilaisia lisenssejä. Lisenssien tarkoitus on suojata tekijänoikeuksia, rajoittaa ohjelmiston käyttöä ja levitystä. Avoimen lähdekoodin ohjelmistojen tapauksessa lisenssin tarkoituksena on erityisesti ohjelman levittäminen, mutta lisenssit ovat myös tärkeässä asemassa suojaamassa tekijän ja käyttäjien oikeuksia. Lisenssien merkitys on hyvä ymmärtää, mikäli aikoo käyttää tai kehittää ohjelmistoja. Suurin periaatteellinen ero avoimien ja suljettujen ohjelmien lisensseissä on, että Open Software Initiativen (OSI) mukaan avoimen lähdekoodin lisenssien tulee täyttää OSI:n avoimen lähdekoodin määritelmän ehdot, jotka käytiin läpi edellisessä kohdassa. OSI-järjestö on standardoinut kymmeniä lisenssiä, jotka täyttävät tämän määritelmän. Seuraavassa esitellään muutamia lisenssejä.

Public Domain

Tarkoitetaan yleisomistukseen tehtyjä ohjelmistoja ja ohjelma on vapaasti käytettävissä mihin tahansa tarkoitukseen. Ohjelman tekijä on luopunut tekijänoikeuksistaan. Lähdekoodi on vapaasti käytettävissä.

Shareware

Ohjelmiston käyttö on aluksi ilmaista, mutta jossain vaiheessa sen käytöstä pitää maksaa lisenssimaksu. Shareware-ohjelmiston tarkoituksena on lähinnä tutustuttaa ihmisiä ohjelmaan ja saada se mahdollisimman monen saataville. Shareware-ohjelmistoa käytetään usein markkinointitarkoituksiin. Lähdekoodi ei ole vapaasti saatavilla.

Freeware

Ohjelmiston käyttö on ilmaista. Freeware-ohjelmistoa käytetään myös usein markkinointistrategiana mainostamaan täydentävää ohjelmaa, jolloin ilmainen osa voi olla karsittu alkuperäisestä. Lähdekoodi ei ole vapaasti käytettävissä.

GNU General Public License, GPL

GPL-lisenssi on avoimen lähdekoodin yleisin lisenssi. Lisenssin perusajatus on yksinkertainen, jokainen saa kopioida, levittää ohjelmaa ja muuttaa ohjelmakoodia tämän GPL-lisenssin alaisuudessa. Lisenssillä on myös ehtoja: lähdekoodin oltava helposti käyttäjien saatavilla, lisenssi pitää sisältyä levitettävään ohjelmaan ja vuorovaikutteisissa ohjelmissa tulee näyttää tiedote käynnistäessä. Kun tehdään muutoksia ohjelmaan, pitää ilmoittaa mitä muutoksia on tehty, milloin ne on tehty ja kuka ne on tehnyt. Tunnetuimpia GPL-lisenssin esimerkkejä ovat GNU-projekti ja Linux-käyttöjärjestelmä.

GNU Lesser General Public License, LGPL

Koska GPL-lisenssin kaupallinen käyttö on rajoitettua, Free Software Foundation kehitti LGPL-lisenssin. Se sallii kaupallisten ohjelmien käyttää kirjastoja.

Mozilla Public License, MPL

MPL-lisenssi on Netscapen muunnelmä GPL-lisenssistä. Tämän lisenssin alaisuudessa se julkaisi oman internet-selaimensa Mozillan lähdekoodin. MPL-lisenssi sallii ohjelmien sisältää myös sellaisia osia, joiden lähdekoodia ei ole pakko julkaista. MPL-lisenssin ongelmana on yhteensopimattomuus GNU-lisenssien kanssa.

BSD lisenssi

BSD tyylliset lisenssit ovat vanhimpia avoimen lähdekoodin lisenssejä. BSD-lisenssin alaisen ohjelman levittämien ja käyttö on sallittu seuraavin ehdoin:

- Lähdekoodikopioissa on oltava maininta tekijänoikeudesta, nämä ehdot ja seuraava vastuunrajoitus dokumentaatiossa tai muussa mukana tulevassa aineistossa;
- Ohjelmiston kehittäjien nimeä ei saa käyttää ohjelmistosta kehiteltyjen tuotteiden mainostamiseen ilman erillistä kirjallista lupaa;

- Tekijänoikeuden haltijat ja ohjelmiston kehittäjät tarjoavat ohjelman sellaisenaan, mitään takuita ei myönnetä;
- Tekijänoikeuden haltija ja ohjelmiston kehittäjä eivät ole vastuussa mistään ohjelmiston käytöstä seuranneista vahingoista.

Taulukko 1. Kooste lisensseistä ja niiden ominaisuuksista.

Lisenssi	Saata- villa ilmai- seksi	Levittä- minen sallittu	Ei käyttö- rajoit- uksia	Lähde- koodi vapaasti saata- villa	Lähde- koodin muokkaa- minen sallittu	Johdan- naisen työn oltava vapaa	Sidos patentoituun ohjelmistoon sallittu
Public Domain	X	X	X	X	X		X
Shareware	X *	X					
Freeware	X	X	X				
GPL	X	X	X	X	X	X	
LGPL	X	X	X	X	X	X	X
MPL	X	X	X	X	X	X	X
BSD	X	X	X	X	X		X
*Shareware on ilmainen vain koeajan							

Usein ongelmaksi kehittyy useiden eri tahojen tekemät lisenssit. Tällöin dokumentaatioissa pitää olla kaikkien tekijänoikeuksien haltijoiden versio lisensseistä ja heidän tekijänoikeuksistaan. Jokainen projekti käyttää omaa versiota lisensseistä, joten mitään yksittäistä lisenssiä ei ole. Usein ei ole tarve kuitenkaan muuttaa muuta kuin tekijänoikeuksien omistajaan liittyvät kohdat.

[17; 18]

7.3 Avoimen lähdekoodin ohjelmistojen historiaa

Avoimen lähdekoodin ohjelmistojen kehitys alkoi jo 1970-luvulla, sillä ei ollut selkeää nimeä ja se ei ollut todellinen vaihtoehto. Se tunnettiin nimellä vapaa ohjelmisto. Vapaa ohjelmisto ilmestyi jo ennen, kuin ihmiset alkoivat miettiä, kuka omistaa ohjelmistot. Tässä tutkimuksessa keskitytään pääasiassa avoimen lähdekoodin ohjelmistoihin, koska vapaalla ohjelmistolla ja avoimen lähdekoodin ohjelmistolla ei ole huomattavaa eroa tämän tutkimuksen näkökulmasta.

Avoimen lähdekoodin historiassa ohjelmien kehittäjät kohtasivat monia ongelmia, koska he eivät saaneet käyttöönsä lähdekoodia. He halusivat kehittää ohjelmia ja korjata ohjelmissa olevia virheitä. Kehittäjät eivät olleet tyytyväisiä kaupalliin ohjelmiin. Heidän mielestään ohjelmista puuttui luovuutta ja vapaata kehitystä. Tämän seurauksena he päättivät tehdä ohjelmat itse alusta alkaen.

Vuonna 1983 Richard Stallman aloitti GNU-projektin, joka keräsi samoin ajattelevia ohjelmistokehittäjiä yhteen. He alkoivat kehittää vapaita ohjelmistoja. Vuonna 1985 Richard Stallman julkaisi GNU-manifestin, jossa hän ilmaisi näkemyksensä tekijänoikeuksista ohjelmistojen osalta. Richard Stallmannin osuus on vapaan ohjelmiston ja avoimen lähdekoodin ohjelmistojen kehityksessä on merkittävä. Hän oli mukana perustamassa Free Software Foundation (FSF) järjestöä. Järjestö tuki vapaan ohjelmiston kehittämistä ja se myös määritteli termin vapaa ohjelmisto virallisesti. Vapaa ohjelmisto termi määrittelee oikeudellisia ja käsitteellisiä periaatteita.

Vuonna 1991 Linus Torvalds, 21-vuotias suomalainen ohjelmoija, aloitti kuuluisan Linux-projektin, joka on julkaistu GPL-lisensillä. Tämä toteuttaa Unixin kaltaisen käyttöjärjestelmän Intel-pohjaisille mikrotietokoneille. Projekti on kasvanut nopeasti ja tuottanut nopean, tehokkaan, vakaan, luotettavan ja skaalautuvan käyttöjärjestelmän. Linuxin käyttäjien ja kehittäjien kasvaa jatkuvasti.

Vuotta 1998 voidaan pitää käännekohtana avoimen lähdekoodin ohjelmistokehityksessä. Bruce Perens ja Eric Raymond toivat ilmi epäilyksensä, että yritykset eivät olleet vakuuttuneita vapaa ohjelmistotermistä, jonka Richard Stallmanin oli määritellyt. Vapaa ohjelmiston perusajatus oli vahvasti liiketoiminnan vastainen. Richard Stallmanin käsitys vapaasta ohjelmasta oli enemmän ideologinen ja filosofinen, kun taas Perensin ja Raymondsin käsitys lähdekoodista oli enemmän liiketoiminnan näkökulmaan suuntautuva. Näiden kahden näkökannan välille muodostui kiulu. Perensin ja Raymondsin käsitys lähdekoodista sai kuitenkin nopeasti enemmän jalansijaa. Näin luotiin avoimen lähdekoodin ohjelmiston liike. Useiden suurten avoimen lähdekoodin ohjelmiston projektien mukana olleitten ihmisten mielissä tämä prosessi oli ohjelmiston vallankumous.

Nykyään kaikki suuret laitteisto- ja ohjelmistovalmistajat ottavat huomioon avoimen lähdekoodin lähestymistavan. Esimerkiksi IBM julkisti vuonna 2005 suunnitelmansa kuluttaa 100.000.000 dollaria kolmen seuraavan vuoden aikana rakentaen Linux-tukea sen sovelluksiin.

Nykyisin on olemassa myös kaupallinen avoin lähdekoodi, jolle on tyypillistä avoimen lähdekoodin hyödyntäminen osana perinteisesti lisensoituja kaupallisia tuotteita, palveluita ja alan vakiintuneita liiketoimintamalleja. Avoimen lähdekoodin ohjelmistojen avulla pyritään erityisesti pienentämään kustannuksia ja lisäämään kehitysnopeutta. Tyypillisiä ovat suljetun ja avoimen lähdekoodin teknologioita yhdistävät tuotteet, joihin sisältyy sekä avoimia että suljettuja komponentteja. Tämä kehitysvaihe alkoi noin vuonna 2005.

[14; 16; 18]

7.4 Ohjelmistokehitys

Avoimen lähdekoodin ohjelmien kehitys eroaa kaupallisten ohjelmistojen kehittämisestä sillä, että hyvin usein projektit etenevät yleensä harrastuspohjalta. Myös avoimen lähdekoodin ohjelmia tehdään kaupallisesti, mutta yleensä niiden pohjalla ei ole kaupallista organisaatiota. Avoimen lähdekoodin projektit lähtevät yleensä liikkeelle hitaasti, koska kehittäjät tekevät ohjelmaa yleensä vapaa-ajallaan. Kun kiinnostus projekti kohtaan kasvaa, tulee uusia kehittäjiä ja projekti nopeutuu. Avoimen lähdekoodin ohjelmien kehittäminen on jatkuvaa iterointia ja testausta.

Ohjelmistokehitysmallit katedraali ja basaari

Kaksi yleisimmin käytössä olevaa ohjelmistokehitysmallia ovat katedraali ja basaari. Katedraalimalli perustuu keskitettyyn suunnitteluun ja toteutukseen ja on lähellä perinteistä ohjelmistotuotannon mallia. Projektin kehittävät lähettävät ehdotuksiaan päättäjäjoukolle, jotka tekevät päätöksen uusista ominaisuuksista tai korjauksista. Basaarimalli perustuu hajautettuun suunnitteluun ja toteutukseen. Siinä kaikilla ryhmään valituilla kehittäjillä on oikeus viedä projektia eteenpäin. Basaarimallissa eri versioiden julkaisutiheys saattaa olla hyvinkin suuri, jopa päivittäin. Avoimen lähdekoodin projekteissa basaarimalli on yleisempi.

Ohjelmiston kehittäjät

Avoimen lähdekoodin ohjelmistojen kehittäjiä on monelta eri alalta, enimmäkseen yliopistoista, julkiselta sektorilta, yrityksistä ja lisäksi on yksityisiä henkilöitä. Yliopistoilla on lisäksi myönteinen suhtautuminen avoimen lähdekoodin ohjelmiin ja niiden lisensseihin, mikä mahdollistaa myös niiden levittämisen laajemmalle.

Osalla avoimen lähdekoodin ohjelmistojen jakelijoista on myös merkittävä osuus sen kehittämisessä. Jakelijat saattavat kehittää ohjelmistoa käyttäjäystävällisemmäksi ja paremman näköiseksi. Usein he tekevät myös erilaisia ohjekirjoja ja käyttöoppaita ohjelmistoon. Jakelijoiden motiivina on saada ohjelmisto näyttämään omalta ja ja saada siitä samalla hieman tuottoa.

Jakelijoiden lisäksi myös kaupalliset ohjelmistotalot ottavat osaa kehitykseen. Muun muassa isot ohjelmistotalot HP, IBM ja Oracle käyttävät hyvin paljon resursseja avoimen lähdekoodin kehitykseen. Nämä kolme yritystä myös kauppaavat Linux-käyttöjärjestelmää omien tuotteidensa ohessa ja ovat kiinnostuneita sen kehittämistä.

Ohjelmistokehityksen motiiveja

Avoimen lähdekoodin ohjelmistojen kehitystyössä on harvemmin motiivina taloudelliset syyt. Usein on kysymyksessä kehitystyön tuloksesta saatava hyöty omaan käyttöön. Monissa tutkimuksissa on käynyt myös ilmi, että suurin motiivin lähde on kilvoittelu. Kehittäjät auttavat toisiaan ja opettelevat muiden kehittäjien taitoja. Seuraavassa luetellaan motiiveja avoimen lähdekoodin ohjelmistokehitykselle:

- Kehittäjä haluaa menestyä urallaan, saada julkisuutta ja mahdollisesti sitä kautta paremmin palkattua työtä;
- Kehittäjä nauttii luovasta ohjelmoinnista ja ohjelmiston kehittämisestä. Hän haluaa kehittää häntä itseään kiinnostavia ohjelmistoja;
- Kehittäjä toimii idealogisin perustein. Hän tukee ajatusta, että kaikki ohjelmat pitäisi olla avoimia ja paheksuu kaupallisten ohjelmien toimintatapaa, jossa ohjelmia kontrolloidaan kaupallisin perustein;
- Monet yritykset ovat mukana taloudellisin perustein avoimen lähdekoodin ohjelmien tuottamisessa ja kehittämisessä;
- Tapauksissa, joihin ei ole sopivaa ohjelmistoa valmiina, esimerkiksi koulutuksessa ja tutkimustyössä. Uusi ohjelmisto kehitetään ensin omiin tarpeisiin, jonka jälkeen se julkaistaan avoimen lähdekoodin ohjelmistona.

Monet avoimen lähdekoodin ohjelmistojen kehittäjistä ovat ammatti-kehittäjiä ja he työskentelevät kaupallisissa yrityksissä. Heidän kehitysmallinsa on yleensä basaarimalli, sen antaman vapauden takia.

[17]

7.5 Liiketoiminnan näkökulma

Avoimen lähdekoodin ohjelmien kehitystyö on usein rahoitettu projektista kiinnostuneiden osapuolten toimesta. Kiinnostuksen takana on monesti ideologiset, poliittiset ja kaupalliset syyt. Yksi merkittävä tulon lähde on julkisen sektorin avustukset, jonka tutkimusrahoista välittyy myös osa avoimen lähdekoodin ohjelmien kehittämiseen. Myös suuret ja merkittävät yritykset kuuluvat tukijoihin. Esimerkiksi Mozilla Firefox internet-selaimen kehittäjät saivat avustuksia Googlelta ja muilta yrityksiltä.

Seuraavassa esitellään kaupallisia malleja, jotka perustuvat avoimen lähdekoodin ohjelmiin. Yritykset saattavat toimia suoraan näiden mallien pohjalta tai näiden eri variaatioiden pohjalta.

Ohjelmiston jakelu – avoimen lähdekoodin ohjelmistoon perustuvan paketin myynti

Yritykset ovat huomanneet, että osaa yhteiskunnasta on alkanut kiinnostaa avoimen lähdekoodin ohjelmistojen tuotteet. Tämän seurauksena yritykset ovat tulleet mukaan kilpailuun tuomalla omia avoimen lähdekoodin ohjelmistoihin kuuluvia tuotteita.

Kehitys ja yhtiön sisäisesti kehitetyn tuotteen myynti

Asiakkaalle tehdään yksilöity sovellus alkuperäisestä ohjelmasta. Sovellus vastaa täysin asiakkaan tarpeita. Se ei sisällä mitään ylimääräisiä sovelluksia, jotka nostaisivat kokonaishintaa.

Lisäarvomyyynnit – avoimen lähdekoodin ohjelmistoja on käytetty edistämään omien sovellusten tai laitteistojen myyntiä

Kaupallisen ohjelman rinnalle julkaistu avoimen lähdekoodin ohjelmiston sovellus, joka sisältää rajoitetun määrän ominaisuuksia verrattuna suljettuun ohjelmaan. Tällaisella versiolla asiakas voi tutustua ohjelmaan ja välttyä kalliilta lisenssimaksuilta. Jatkossa hän voi myös hankkia kokonaisen kaupallisen version.

Palvelut – tuki, koulutus ja konsultointi

Yritys tarjoaa asiakkaille erilaisia palveluja: koulutusta, konsultointia, tuotteen räätälöintiä ja käytön tukea. Asiakasta veloitetaan ainoastaan näistä palveluista. Tämän mallin varjopuolena on, että kilpaileva yritys voi helposti tuottaa samat palvelut alemmalla hinnalla.

Oheistuotteet – kirjallisuus, manuaalit

Jotkut avoimen lähdekoodin ohjelmistojen valmistajat saattavat tarjota ohjelmistoa hyvin edullisesti tai jopa ilmaiseksi, mutta toiminta säilyy kannattavana oheistuotteiden ansiosta.

Vapaamielisempi ja rajoittavampi lisenssi

Käytettäessä avoimen lähdekoodin ohjelmistoja ei yleensä tarvitse miettiä tekijänoikeuksia ja lisenssien eri ominaisuuksia. Lisenssin valinta ja ehdot tulee selvittää, jos aikoo jakaa itse kehitettyä ohjelmaa eteenpäin.

Yleensä valinta tehdään kahden eri lisenssityypin välillä:

- Vapaamielisempi lisenssi, esimerkiksi BSD. Tämän tyyppinen lisenssi mahdollistaa kaupalliset tarkoitukset lähdekoodin käytössä.
- Rajoittavampi lisenssi, esimerkiksi GPL. Tämän tyyppinen lisenssi takaa ohjelmiston säilymisen avoimena, eikä sitä voida missään vaiheessa käyttää kaupallisesti.

Molemmissa vaihtoehtoissa on omat hyvät ja huonot puolensa, joten avoimen lähdekoodin ohjelmistoprojekti on hyvä miettiä yksilönä ennen lisenssivalintaa.

[13; 17]

7.6 Avoimen lähdekoodin ohjelmistot vs. suljetut ohjelmistot

Avoimen lähdekoodin ohjelmistojen vastustajat esittävät usein monia väitteitä, joilla yrittävät saada aikaan kielteistä suhtautumista aiheeseen. Toisaalta avoimen lähdekoodin ohjelmien puolestapuhujat yrittävät saada aikaan kielteistä suhtautumista kaupallisiin, suljettuihin ohjelmiin.

Yksi yleinen harhaluulo on, että virukset ja muut haittaohjelmat leviävät myös Linuxiin, jos sen käyttö yleistyy. Näin ei kuitenkaan ole, koska Linuxin arkkitehtuuri on toteutettu

aivan eri tavalla kuin Windows. On kuitenkin huomioitava, että myös Linuxia vastaan on viruksia liikkeellä, mutta niiden lukumäärä on aivan eri luokkaa kuin Windows-ympäristössä.

Palvelimissa ja mobiililaitteissa Linux on hyvin yleinen käyttöjärjestelmä, koska siitä ei ole löydetty tietoturva-aukkoja läheskään niin paljon kuin Windows-käyttöjärjestelmistä. Linuxista löydetyt tietoturva-aukot on saatu nopeasti korjattua, koska lähdekoodiin on vapaa pääsy ja testaajia on paljon.

Usein esille nostettu asia on, että avoimen lähdekoodin ohjelmien käyttäminen olisi riskialtista, koska kehittäjät eivät ota vastuuta ohjelmistoista seuraavista haitoista. Tosiasia kuitenkin on, että vain harvat yritykset ottavat vastuun ohjelmistojen käytöstä seuraavista haitoista. Niiden yritysten, jotka korvaavat näitä haittoja, korvausmäärät ovat yleensä hyvin alhaisia.

Käyttäjien mielestä etuja avoimen lähdekoodin ohjelmista ovat muun muassa suurempi vakaus, korkea turvallisuuden taso, lisenssimaksujen pienuus, lähdekoodin muokattavuus, mahdollisuus kehittyä paremmaksi ohjelmoijaksi ja riippumattomuus suurista ohjelmistoyrityksistä.

Avoimen lähdekoodin ohjelmien mukanaan tuomat hyödyt:

- Käyttäjä voi itse korjata ohjelmassa olevia virheitä, eikä ole riippuvainen ohjelmiston valmistajasta;
- Monen samankaltaisten ohjelman kehittäminen eri valmistajilla syö turhaan resursseja – keskittämällä kaikki resurssit yhteen ohjelmaan saataisiin parempi tulos;
- Käyttäjä voi muokata lähdekoodista omaan tarkoitukseen yksilöllisen ohjelmiston;
- Ohjelmistoa on mahdollista testata ja kehittää paremmin, koska ympäri maailman on testaajia ja kehittäjiä runsaasti;
- Avoimen lähdekoodin ohjelmistojen kehityksessä päätavoite on laadukkaan ja turvallisen ohjelman tuottaminen, eikä markkinointi ja kaupallisuus;
- Avoimen lähdekoodin ohjelmat ovat jossain määrin laadukkaimpia, sillä niitä kehittävät useimmiten alan parhaimmat asiantuntijat, toisin kuin kaupallisia ohjelmia. Projektiluonteisen kehitysmenetelmän ansiosta huonot tuotokset saadaan poistettua;

- Kokonaiskustannukset ovat alhaisemmat, koska asiakkaan ei tarvitse maksaa tekijänoikeuksista, vaan kulut syntyvät ainoastaan itse kehitystyöstä ja tuesta;
- Ohjelman turvallisuus on parempi, koska lähdekoodista on suoraan nähtävissä tahattomat ja tahalliset virheet. Havaitut virheet voidaan korjata yleensä hyvin nopeasti.

Avoimen lähdekoodin ohjelmien huonot puolet:

- Kaupallisiin ohjelmistoihin verrattuna avoimen lähdekoodin ohjelmistojen tukipalvelu on yleensä heikompaa, koska ohjelmistot ovat enimmäkseen yksittäisten henkilöiden kehittämiä ja ylläpitämiä;
- Avoimen lähdekoodin ohjelmistojen taustalla on harvoin kaupallista markkinointia, joten tukeenkaan ei ole resursseja;
- Ohjelmistojen kehitys on nopeaa ja kehittäjät eivät innoissaan aina osaa huomioida vanhempien versioiden toimivuutta uuden version kanssa. Päivitysväli on lyhyt, joten se vaatii asiakkaalta jatkuvaa tilanneseurantaa;
- Yhteensopivuusongelmia voi odottaa suljettujen ohjelmien kanssa;
- Avoimen lähdekoodin ohjelmien tuottajilla ei ole mitään vastuuta ohjelmiensa aiheuttamista haitoista.

Vaikka siirtyminen kaupallisista ohjelmista avoimen lähdekoodin ohjelmien käyttöön tietääkin korkeita kuluja, monet tutkimukset ovat osoittaneet, että sekä lyhyellä että pitkällä aikavälillä alhaisten käyttökulujen takia kokonaiskustannukset jäävät alhaisemmiksi.

Avoimen lähdekoodin ohjelmia pidetään yleisesti turvallisina, koska lähdekoodi on kaikkien nähtävissä ja korjattavissa. Havaitut virheet ilmoitetaan avoimesti, jolloin ohjelmat voidaan korjata nopeasti. Avoimen lähdekoodin projektin ollessa alkuvaiheessa sen avoimuudesta ei saada kaikkea irti. Kun avoimen lähdekoodin projekti on saavuttanut tietyn toimivuuden ja käyttäjäkunnan, voidaan odottaa korjauksia useammin. Kaupallisissa ohjelmissa pitää vain luottaa, että ohjelmiston valmistaja on valmistanut turvallisen tuotteen. Tosi asia kuitenkin on, että täysin turvallista tuotetta ei ole saatu kehitettyä, oli sitten kysymyksessä avoin tai suljettu ohjelmisto.

7.7 Avoimen lähdekoodin ohjelmistojen tulevaisuus

Avoimen lähdekoodin ohjelmistojen merkitys ohjelmistotuotannossa on voimakkaassa kasvussa ja tätä ilmiötä on viime aikoina tutkittu paljon. Avoimen lähdekoodin ohjelmistoille on puolustajia ja vastustajia lähes yhtä paljon. Avoimen ja suljetun lähdekoodin ohjelmissa on omat hyvät ja huonot puolensa. Siksi ensin pitää selvittää oma käyttötarkoitus ja vaatimukset, tämän jälkeen voi tehdä perusteltuja vertailuja vaihtoehtojen välillä.

Avoimen lähdekoodin ohjelmien tulevaisuudesta on monia arvailuja ja veikkauksia, mutta suurin osa arvostetuista kommentoijista on sitä mieltä, että ilmiö ei ole saavuttanut vielä lopullista tilaansa. Lisäksi ohjelmistot ovat nykyisin kalliita, joten on todennäköistä, että avoimen lähdekoodin ohjelmien suosio tulee laskemaan kaupallisten ohjelmien hintoja.

Avoimen lähdekoodin ohjelmistot tuskin koskaan tulevat syrjäyttämään kaupallisia ohjelmistoja, todennäköistä on, että nämä kaksi ohjelmistotuotannon tapaa tulevat jatkossakin jatkamaan keskinäistä kilpailuaan. Tämän seurauksena ohjelmistot kehittyvät paremmiksi ja kilpailu pakottaa uusiin innovatiivisiin ratkaisuihin.

[17]

8 ERP-JÄRJESTELMÄN VALINTAKRITEERIT AVOIMEN LÄHDEKOODIN NÄKÖKULMASTA

Tosiasia on, että on suuri kiinnostus avoimen lähdekoodin ohjelmistoihin, mutta mitä voidaan sanoa avoimen lähdekoodin ERP-järjestelmistä. Seuraavassa esitellään avoimen lähdekoodin toiminnanohjausjärjestelmän valintakriteereitä tarkeysjärjestyksessä B. Johanssonin ja F. Sudzinan tekemän tutkimuksen mukaan. [19]

Yksinkertainen ja nopea implementointi

On olemassa näkemys, että ERP-järjestelmän implementointi kestää liian kauan. On vaikea sanoa, onko suljetun tai avoimen lähdekoodin ERP-järjestelmät nopeampia ja helpompia toteuttaa. Suljettu ERP-järjestelmä tarkoittaa, että käyttäjällä ei ole oikeutta ohjelman lähdekoodiin, joka on vastakohta avoimen lähdekoodin järjestelmään. ERP-järjestelmän toimittajat tai heidän yhteistyökumppaninsa käyttöönottavat suljetun ERP-järjestelmän, joten ei ole selvää tarvetta tehdä ERP-järjestelmien asennusta helpoksi. Kuitenkin voidaan todeta, että jos ERP-järjestelmä olisi helpompi toteuttaa, silloin ERP-järjestelmien toimittajien olisi mahdollista myydä enemmän lisenssejä ja tällöin myös konsultteja olisi helpommin saatavissa. Lisäksi helppo ja nopea ERP-järjestelmän implementointi ei riipu ainoastaan graafisesta käyttöliittymästä ja asetuksien lukumäärästä, vaan se on riippuvainen ohjelmistomuutoksista ja konfiguroinnista.

Hinta

Avoimen lähdekoodin ERP-järjestelmän implementointi ei ole ilmainen, vaikka kalliita ohjelmiston lisenssimaksuja ei ole. Yrityksen pitää maksaa konsultoinnista, koulutuksesta, ylläpidosta ja tarvittavista räätälöinneistä järjestelmän implementoinnissa. Avoimen lähdekoodin ERP-järjestelmän konsultit voivat veloittaa huomattavasti enemmän, kuin suljetun ERP-järjestelmän konsultit, koska niitä on vähemmän tarjolla. Jos yritykselle on omaa IT-osaamista, se alentaa järjestelmän implementoinnin kustannuksia. Avoimen lähdekoodin ERP-järjestelmä voi olla hyvä ratkaisu aloittavalle yritykselle, joka ei ole vielä vakiinnuttanut liiketoimintaprosessejaan, joten yritys voi sopeutua järjestelmään helpommin.

Toimittajan tuki

Tämä kriteeri on tärkeämpi suljetuissa ERP-järjestelmissä, koska sen toimittajat tarjoavat omille asiakkailleen tukea ja toimittajat tuntevat asiakkaan ERP-järjestelmän. Asiakkaat ovat kuitenkin sidottuja toimittajiin tai heidän yhteistyökumppaneihinsa. Vastaavasti avoimen lähdekoodin ERP-järjestelmän implementointia tukevat avoimen lähdekoodin yhteisöt ja erilaisia tukipalveluita on saatavilla riippuen järjestelmästä.

Luotettavuus

Suljetun ERP-järjestelmien valmistajat yrittävät saavuttaa mahdollisimman korkean luotettavuuden. Suljetun ERP-järjestelmän tekijöiden on kuitenkin vaikeampaa havaita virheitä, koska samat ihmiset tarkastavat koodia ja varsinkin siinä tapauksessa, että koodin on kirjoittanut itse. Avoimen lähdekoodin ERP-järjestelmän ohjelmistojen tekijöillä on tässä etu puolellaan, koska avoimen lähdekoodin kehittäjiä ja testaajia voi olla ympäri maailman. Virheiden korjaus on myös nopeaa johtuen vapaasta pääsystä lähdekoodiin ja ohjelmistokehityksen mallista.

Helppokäyttöisyys

On epäselvää, onko suljettua tai avointa ERP-järjestelmää helpompi käyttää. Suljetun ERP-järjestelmän valmistajat yrittävät saavuttaa kilpailuetua tarjoamalla toiminnanohjausjärjestelmiä, jotka ovat helppoja käyttää. On olemassa käsitys, että avoimen lähdekoodin ratkaisu ei ole kovin käyttäjäystävällinen. Avoimen lähdekoodin ERP-järjestelmien helppokäyttöisyyden taso on yleisesti riittävä käyttäjille, jotka ovat tottuneet käyttämään eri ohjelmistoja. Tosiasia on, että avoimen lähdekoodin yhteisöt ja ohjelmoijat kehittävät järjestelmiä koko ajan paremmaksi ja käyttäjäystävällisemmäksi.

Räätälöinti ja konfigurointi

Suljetut ERP-järjestelmät ovat usein ohjelmoitu tietyn valmistajan ohjelmointikielellä. Yritykset kuitenkin ovat varmoja, että toimittajat voivat räätälöidä ja konfiguroida heidän ERP-järjestelmiään, mutta se tarkoittaa myös sitä, että he ovat sidottu toimittajaan tai heidän yhteistyökumppaniin. Koska avoimen lähdekoodin ERP-järjestelmät ohjelmoidaan laajalti tunnetuilla ohjelmointikielillä, on suurempi potentiaalisten ohjelmoijien joukko, jotka voivat tehdä ohjelmistomuutoksia.

Integraatio

Integraatio toisiin järjestelmiin on usein ongelma asiakkaille. Monet suljetut ERP-järjestelmät mahdollistavat integraation toisiin järjestelmiin jo entuudestaan. Monissa ERP-järjestelmissä, sekä suljetuissa että avoimissa on käytössä API-liitäntä. API-liitäntä

(application programming interface) on ohjelmistorajapinta toiseen ohjelmaan, tämä tiedonsiirtoliitäntä löytyy useista ohjelmistoista. Räätelöity integraatoratkaisu on myös mahdollinen, varsinkin jos järjestelmä pitää liittää vanhempaan ohjelmaan.

Organisatorinen sovitus

Nykyiset ERP-järjestelmät pyrkivät keskittymään liiketoiminnan prosesseihin. On kuitenkin havaittavissa, että ne eivät yksiselitteisesti kuvaa liiketoimintaprosesseja, joita ne tukevat. Sen lisäksi, ERP-järjestelmät eivät ole helposti sovitettavissa liiketoimintaprosessien muutoksiin. Valitettavasti näyttää siltä, että tämä ongelma on sekä avoimen että suljetun lähdekoodin toiminnanohjausjärjestelmissä.

Toiminnallisuus

Mitä tulee toiminnanohjausjärjestelmien toiminnallisuuteen niin suljetut ERP-järjestelmät tarjoavat enemmän vaihtoehtoja, johtuen lähinnä siitä, että toimittajat ovat olleet kauemman alalla. Toisaalta taas avoimen lähdekoodin ERP-järjestelmiä, koska ne tarjoavat vähemmän vaihtoehtoja, on helpompi hallita ja toteuttaa, kuin suljettua toiminnanohjausjärjestelmää.

Toimittajan maine

Toimittajan maine voidaan arvioida ERP-järjestelmän laadusta, jos asiakkaalla ei ole mahdollisuutta tutkia ohjelmistoa yksityiskohtaisesti ennen ostopäätöstä. Avoimen lähdekoodin ERP-ohjelmiston voi kuka tahansa ottaa käyttöön ja tutkia, joten tällä kriteerillä ei ole sellaista merkitystä kuin suljetuissa ERP-järjestelmissä.

Ohjelmiston joustavuus

Ohjelmiston joustavuus on tarpeen, johtuen nopeasti muuttuvasta liiketoimintaympäristöstä, mikä todennäköisesti johtaa uusiin liiketoimintamalleihin. Liiketoimintamalleja on tuettava häiritsemättä käynnissä olevaa liiketoimintaa ja ERP-järjestelmän käyttäjiä. Tämä edellyttää toiminnanohjausjärjestelmältä vankkaa arkkitehtuuria, joka mahdollistaa liiketoimintamallien muutokset. Monet nykyiset suljetut ERP-järjestelmät saivat alkunsa MRP-järjestelmistä pienin iteraation, joten yleensä niissä ei ole otettu huomioon liiketoimintamallien muutosta. Näin ollen järjestelmät, jotka on kehitetty viime aikoina ja joissa ei ole tätä rasiitetta, ovat todennäköisemmin joustavampia. Avoimen lähdekoodin ERP-järjestelmissä on yleensä otettu huomioon liiketoimintamallien muutokset.

Koulutus

Ensinnäkin voidaan todeta, että vaadittavan koulutuksen määrä riippuu siitä kuinka monia eri toimintoja ERP-järjestelmä tarjoaa ja kuinka looginen ja helppokäyttöinen se on. Toiseksi suljetun ERP-järjestelmän käyttäjät ovat enemmän tai vähemmän riippuvaisia toimittajien tai kumppaneiden järjestämästä koulutuksesta. Toisaalta avoimen lähdekoodin ERP-järjestelmiin voi olla vaikea löytää konsulttiyritystä, joka tarjoaa koulutusta kyseiseen järjestelmään. Koulutus voidaan järjestää myös organisaation omilla resursseilla, jos yrityksessä on kiinnostuneita henkilöitä perehtymään avoimen lähdekoodin ERP-järjestelmään.

Raportointiominaisuudet

Yrityksellä on suuret odotukset uudesta ERP-järjestelmästä, koskien nimenomaan sitä millaista tietoa sieltä on saatavissa. Todennäköisesti kaikki ERP-järjestelmät parantavat informaation laatua. Yritykset ovat nimenomaan kiinnostuneita hyvistä ja helppokäyttöisistä raportointi ominaisuuksista. Yritykset tarvitsevat usein raportteja, joita ei ole sisällytetty ERP-järjestelmään, johtuen lähinnä siitä, että eri yrityksillä monia erityistarpeita, joita on mahdotonta ennakoida. Tässä tapauksessa avoimen lähdekoodin ERP-järjestelmät voivat osoittautua paremmiksi, koska sen ohjelmistoa on helpompi muokata. Avoimen lähdekoodin ERP-järjestelmän taustalla olevaan tietokantaan ja siellä oleviin tietokantatauluihin voidaan lisätä ylimääräisiä sarakkeita, tämä vaatii kuitenkin riittävästi taitoa tietokanta- ja ohjelmisto muutoksista.

Viimeisin teknologia

Usein suljetun ERP-järjestelmien myyjät väittävät, että heillä on käytössään uusin teknologia. Tällöin heillä olisi parempi kilpailuasema toisiin toimittajiin. Avoimen lähdekoodin ERP-järjestelmien kehittäjien käytössä on aivan vastaava teknologia. Sekä suljetun että avoimen ERP-järjestelmän kehittäjät voivat edistää uuden teknologian käyttöä.

Skaalautuvuus

Skaalautuvuus on erityisen tärkeä yrityksille, jotka suunnittelevat yritykselle huomattavaa kasvua. Valitettavasti on monia, sekä avoimia että suljettuja ERP-järjestelmiä, jotka on optimoitu tai ei ole optimoitu ollenkaan tietyn kokoisiin organisaatioihin. Jopa niin tunnettu toimittaja kuin SAP ei kyennyt tekemään ERP-järjestelmää, joka sopii erikokoisiin organisaatioihin. SAP on maailman suurimpia ohjelmistoyrityksiä, joka on erikoistunut ERP-järjestelmiin.

Päivitykset

Päivitykset avoimen lähdekoodin ERP-järjestelmiin ovat ilmaisia. Suljettuihin ERP-järjestelmiin asiakkaan pitää ostaa päivityksiä tai maksaa vuosimaksua, jolloin he saavat päivitykset ilmaiseksi. Tosiasia on, että on olemassa suhde ohjelmistomuutoksien ja päivityksen kustannuksilla, jotka ovat hyvin tehty. Suljetun ERP-järjestelmän toimittajat miettivät tarkemmin virheen korjauksia ja muutoksia, jotka todennäköisesti saattaisivat vaikuttaa ohjelmiston päivitykseen. Pahimmassa tapauksessa päivitys voi vaatia uudelleenohjelmointia moniin aikaisempaan versioihin. Kun ohjelmistovirhe korjataan johonkin päivitykseen tai versioon on todennäköistä, että myös muut yritykset haluavat vastaavat korjaukset omiin ohjelmiinsa.

Liittyen tukeen, avoimen lähdekoodin ERP-järjestelmän käyttäjät asentavat aina uusimman version ja ainoastaan uusin versio on tuettu. Suljetun ERP-järjestelmän toimittajat tukevat aina useita versioita, joissakin tapauksissa kaikkia ERP-järjestelmän versioita. Päivitykset avoimen lähdekoodin ERP-järjestelmään tulevat, kun virhe on löydetty tai joku kehittää ohjelmaan uusia ominaisuuksia. Suljetun ERP-järjestelmän toimittajat yrittävät minimoida päivityksien määrää, sillä monien versioiden tukeminen on kallista.

Modulaarisuus

Mitä tulee modulaarisuuteen, on olemassa ääriesimerkkejä kiinteistä ja modulaarisista ohjelmista sekä avoimen että suljetun lähdekoodin toiminnanohjausjärjestelmissä. Sopiva arkkitehtuuri voi sallia täysin uuden moduulin ohjelmoinnin sekä avoimen että suljetun lähdekoodin toiminnanohjausjärjestelmissä. Mitä tulee varsinaiseen ERP-järjestelmän moduulien saatavuuteen, niitä on luultavasti eniten SAP-toimittajan ERP-järjestelmässä.

Yhteenveto valintakriteereistä

Avoimen lähdekoodin toiminnanohjausjärjestelmät ovat yhä kiinnostavampia tällä hetkellä. Voidaan miettiä, onko tämä kiinnostus seurausta tyytymättömyydestä suljettuun ERP-järjestelmään vai onko se seurausta siitä, että avoimen lähdekoodin ohjelmat ovat olleet jo pitkään esillä. Valintaperusteiden perusteella ei voida päätellä, onko avoimen tai suljetun lähdekoodin ERP-järjestelmien ratkaisut enemmän tai vähemmän sopivia useimmissa organisaatioissa.

Palataan valintakriteereihin, mitkä ovat syyt avoimen lähdekoodin ERP-järjestelmän käyttöön yrityksessä. Selvästi erottuvin kriteeri on hinta, joka tulee toiseksi. Korkein kriteeri on yksinkertainen ja nopea implementointi; voidaan arvioida, että tämä kriteeri vaihtelee paljon näiden kahden vaihtoehdon välillä. Avoimen lähdekoodin kannattajien

mukaan ERP-järjestelmä voidaan vain yksinkertaisesti 'ladata' ja ottaa käyttöön. Toimittajan tuki ja luotettavuus ovat sijoilla kolme ja neljä; voidaan varmasti sanoa, että nämä kaksi perustetta liittyvät toisiinsa. Verrattaessa näitä kahta vaihtoehtoa, avoimen ja suljetun vaihtoehdon välillä, voidaan päätellä, että suljettu ERP-järjestelmä näyttää olevan parempi.

[19]

9 AVOIMEN LÄHDEKOODIN ERP-JÄRJESTELMIÄ

Tässä luvussa luodaan lyhyt katsaus olemassa oleviin järjestelmiin, suosituimpiin. Avoimen lähdekoodin ERP-järjestelmiä on olemassa markkinoilla huomattava määrä, ominaisuuksiltaan ja toiminnoiltaan erilaisia.

Seuraava InfoWorldistä otettu toimittajan kommentti kertoo avoimen lähdekoodin ohjelmistojen valinnan vaikeudesta. "Jos luulet, että voit helposti arvostella ja laittaa paremmuusjärjestykseen avoimen lähdekoodin ohjelmistoja, niin tulet pian huomaamaan, että voit vain repiä hiuksia, pureskella kynsiä ja kiristellä hampaita. On aivan liian paljon hyviä ja erinomaisia työkaluja." [20]

Alla muutamia poimintoja avoimen lähdekoodin ERP-tuotteista.

- Avoimen lähdekoodin konsultti nostaa esiin seuraavat tuotteet: Compiere, OpenBravo, Opentaps, Factualux , WebERP ja OpenERP. [19]
- Seuraavat tuotteet löydetään, pääpaino on pk-yritykset, teollisuusyritykset, Saksan ja Unkarin markkinat: Adempiere, Apache Ofbiz, AvERP, CAO-Faktura, Compiere, Limbas, Lx-Office, Openbravo, OpenERP, Opentaps, SQL-Ledger, Tryton, WebERP ja xTuple ERP. [21]
- Artikkelissaan Serrano ja Sarriegi nostavat esiin seuraavia tuotteita: Compiere, ERP5, Openbravo, Fistera, OFBiz, Value Enterprise, GNU Enterprise, SQL Ledger, WebERP ja OpenERP. [22]
- xTuble ERP palkittiin parhaana Open Source 'Best Business Application' vuonna 2008. [23]
- Palkitut (2009 InfoWorld Bossie Award) Open Source ERP ovat: Compiere ja Openbravo. Muut, jotka mainitaan ovat: OpenERP, Apache OFBiz ja OpenERP. [20]

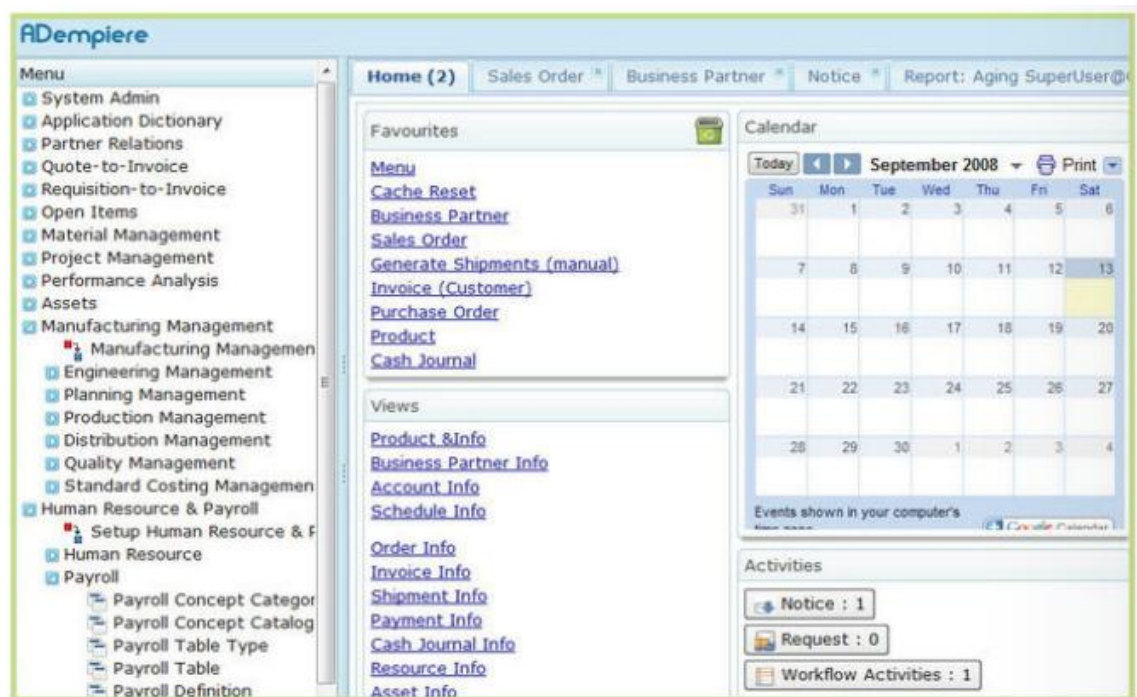
Adempiere <http://adempiere.org/site/>

Adempiere-järjestelmää on kehitetty vuodesta 2006 lähtien. Ohjelmisto on suunniteltu Compiere-järjestelmän pohjalta. Se on ohjelmoitu Java-ohjelmointikielellä ja toimii GPL-lisenssin alaisuudessa. Se toimii Windows- ja Linux-käyttöjärjestelmillä ja sitä voi käyttää selaimella ilman kaupallista lisenssiä toisin kuin Compiere. Järjestelmä on monikielinen. Adempiere tukee myös vahvasti tuotannonohjausta. Se tarjoaa myös tietojen tuonti ja vienti toiminnot.

Modulit ovat myynnin hallinta, ostojen hallinta, tuotannonohjaus, materiaalien hallinta, laskentatoimen hallinta, raportointi, projektinhallinta, tarjousten ja hinnoittelun hallinta, tilausten hallinta, henkilöstöhallinto ja asiakkuudenhallinta eli CRM.

Tukea tarjotaan vain IT-palveluiden tarjoajille. Siihen kuuluu konsultointia, ylläpitoa, räätälöinnin ja integraation toteutusta. Lisäksi on tarjolla kattava verkkopalvelu ja erilaisia oppaita.

Kuvassa 7 on näkymä Adempiere-järjestelmän käyttöliittymästä.



Kuva 7. Näkymä Adempiere-järjestelmän käyttöliittymästä.

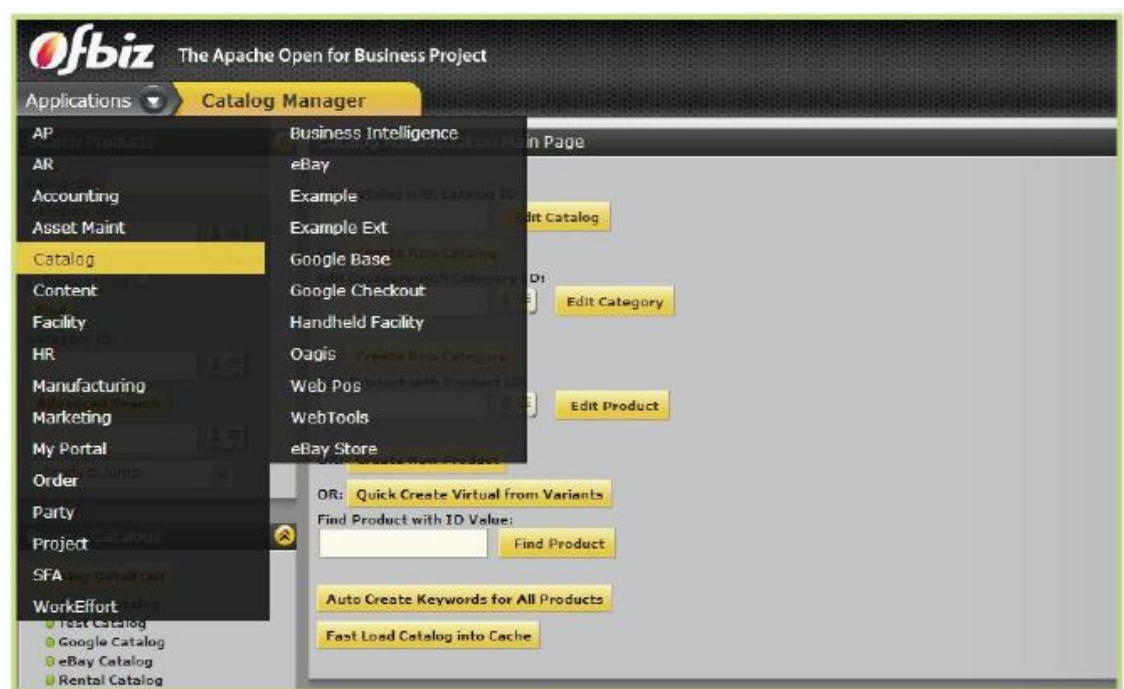
OFBiz <http://ofbiz.apache.org/>

OFBiz-järjestelmää alettiin kehittää vuonna 2001. Se on ohjelmoitu Java-ohjelmointikielellä ja julkaistu Apache-lisenssin alaisuudessa. Se toimii Windows- ja Linux-käyttöjärjestelmillä. OFBiz-järjestelmä tarjoaa myös selainkäyttöliittymän. Graafinen suunnittelu on melko heikkotasoisista ja ohjelmisto ei ole kovin käyttäjäystävällinen. Toisaalta sen hyvänä puolena on erittäin joustava arkkitehtuuri, joka mahdollistaa yksittäisten ohjelmien osien räätälöinnin helposti. Se tukee useita tietokantoja ja tarjoaa myös tietojen tuonti ja vienti toiminnot.

OFBiz-järjestelmän moduulit ovat tilausten hallinta, ostojen hallinta, tuotannonohjaus, materiaalien hallinta, verkkoliiketoiminta, tuoteluettelon hallinta, tarjousten ja hinnoittelun hallinta, asiakkuudenhallinta, varastohallinta, laskentatoimen hallinta ja työkuorman hallinta.

Tuki toimii IT-palvelun tarjoajien kautta. Siihen kuuluu koulutusta ja seminaareja. Ohjelman käyttöön tarjotaan myös ohjeita YouTube-kanavan kautta. Lisäksi on tarjolla verkkopalvelu ja kirjoja.

Kuvassa 8 on näkymä OFBiz-järjestelmän käyttöliittymästä.



Kuva 8. Näkymä OFBiz-järjestelmän käyttöliittymästä.

Compiere <http://www.compiere.com/>

Compiere-järjestelmää on kehitetty vuodesta 1999. Suurin osa järjestelmästä on julkaistu GPL-lisenssin alaisuudessa. Osa järjestelmästä on kuitenkin kaupallisen lisenssin Professional Edition alaisuudessa, kuten selainohjelma ja osa tuotannonohjauksesta. Compiere-järjestelmä toimii Windows- ja Linux-käyttöjärjestelmillä ja on ohjelmoitu Java-ohjelmointikielellä. Compiere-järjestelmä on monikielinen ja tarjoaa myös tietojen tuonti- ja vientitoiminnot. Verkkoyhteyksissä käytetään salausta.

Compiere-järjestelmän moduulit ovat tuotannonohjaus, materiaalien hallinta, laskentatoimen hallinta, ostojen hallinta, projektinhallinta, myynnin hallinta, verkkoliiketoiminta, vahittäismyynnin hallinta ja tulosohtauksen hallinta.

Ohjelmiston kehittäjät ja IT-palveluntarjoajat tukevat ohjelmistoa laajasti.

Kuvassa 9 on näkymä Compiere-järjestelmän käyttöliittymästä.

The screenshot shows the 'COMPARE EDITIONS' page on the Compiere website. The page is viewed in a Mozilla Firefox browser. The URL is <http://www.compiere.com/products/compiere-editions/index.php>. The page has a sidebar on the left with navigation links: Overview, Capabilities, Platform, Global Ready, Cloud Services, Professional Edition, Enterprise Edition, Compare Editions (selected), Open Source, Free Trials & Downloads, and TCO Analysis. The main content area is titled 'COMPARE EDITIONS' and contains a table comparing four editions: Community Edition, Standard Edition, Professional Edition, and Enterprise Edition. The table lists various features and their availability across the editions, marked with green checkmarks. To the right of the table, there are sections for 'Next Steps' (Trials & Downloads, Compare Editions, Contact Us), 'Events' (Online Introductions), 'Related Links' (Buy Compiere Enterprise Edition, Buy Compiere Professional Edition, Buy Compiere Standard Edition, Industry Solutions, Partner Locator), and 'Resources' (Compiere Enterprise Edition Datasheet, Compiere Professional Edition Datasheet, Success Stories, TCO Analysis, Whitepapers & Datasheets, Webcast Archive).

Editions	Community Edition	Standard Edition	Professional Edition	Enterprise Edition
Subscription price (per user, per year)	Free	\$400	\$700	\$995
Minimum number of users	Unlimited	10	10	10
Minimum subscription term	Unlimited	1 year	1 year	1 year
Functionality				
Core ERP & CRM Functionality	✓	✓	✓	✓
Cross-platform Java Client	✓	✓	✓	✓
PDF Report Writer		✓	✓	✓
Web-based Architecture and UI			✓	✓
Business View Reporting Layer			✓	✓
Enterprise-level Data Security Support			✓	✓
Available on Amazon Cloud			✓	✓
Multi-server Support			✓	✓
Management Dashboards			✓	✓
Visual Dictionary Editor			✓	✓
Web Services Support			✓	✓
Compiere Manufacturing			✓	✓
Compiere Warehouse Management			✓	✓
Support				

Kuva 9. Näkymä Compiere-järjestelmän käyttöliittymästä.

[21]

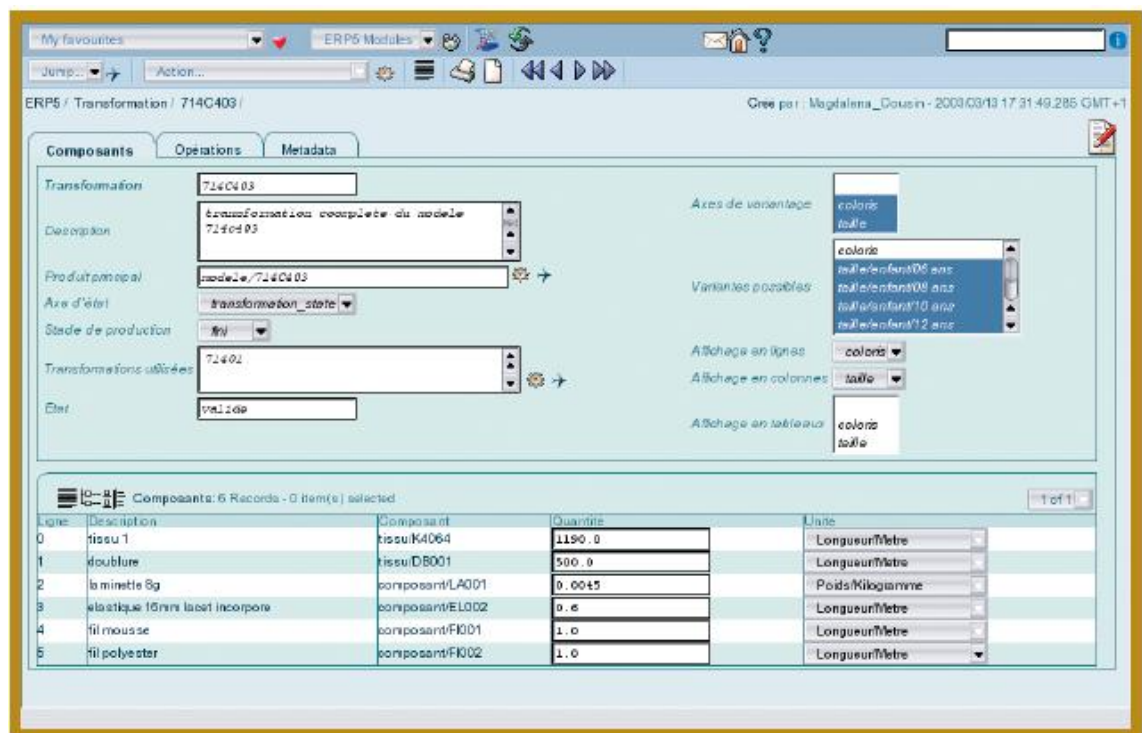
ERP5 <http://www.erp5.com/>

ERP5-järjestelmää alettiin kehittää vuonna 2002. Se on ohjelmoitu Python-ohjelmointikielellä ja julkaistu GNU-lisenssin alaisuudessa. ERP5-järjestelmä on käyttäjäystävällinen ja se toimii Linux-, Windows- ja MacOS- käyttöjärjestelmillä. Verkkoyhteysissä käytetään salausta.

ERP5-järjestelmän moduulit ovat myynnin hallinta, ostojen hallinta, laskentatoimen hallinta, tuotannonohjaus, materiaalien hallinta, varastonhallinta, asiakkuudenhallinta, tuotetiedonhallinta, henkilöstöhallinto, verkkoliiketoiminta ja vähittäismyynnin hallinta.

IT-palveluntarjoajat tukevat ohjelmistoa.

Kuvassa 10 on näkymä ERP5-järjestelmän käyttöliittymästä.



Kuva 10. Näkymä ERP5-järjestelmän käyttöliittymästä.

[22; 24]

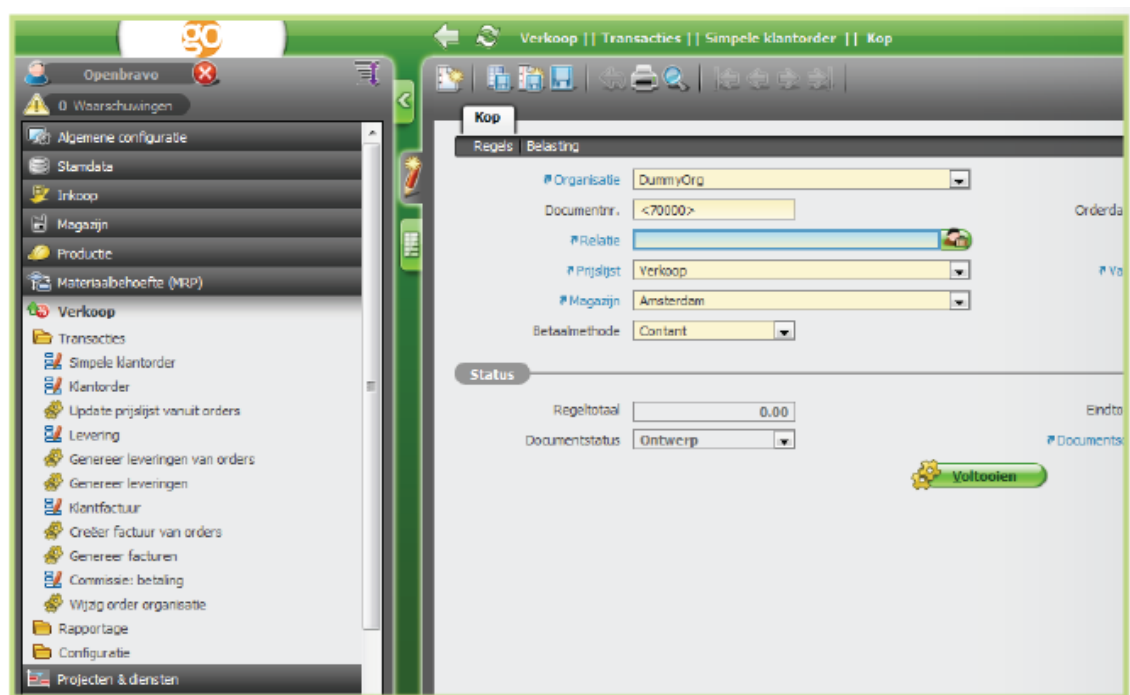
OpenBravo <http://www.openbravo.com/>

OpenBravo-järjestelmää on kehitetty vuodesta 2001. Se on ohjelmoitu Java-ohjelmointikielellä ja toimii Mozilla Public-lisenssin alaisuudessa. Palvelin toimii Linux-, Windows- ja MacOS-käyttöjärjestelmillä. Asiakaspäätte on riippumaton käyttöjärjestelmästä. Se on erittäin käyttäjäystävällinen. Se mahdollistaa myös tietojen tuonnin ja viennin. Verkkoyhteyksissä käytetään salausta. OpenBravo-järjestelmä on monikielinen ja tukee useita valuuttoja.

OpenBravo-järjestelmän moduulit ovat tuotetiedonhallinta, ostojen hallinta, varastonhallinta, projektinhallinta, tuotannonohjaus, materiaalien hallinta, myynnin hallinta, asiakkuudenhallinta, laskentatoimen hallinta ja vähittäismyynnin hallinta.

Kehittäjät ja IT-palveluntarjoajat tarjoavat kattavaa tukea. Käyttäjille ja järjestelmänvalvojille järjestetään seminaareja ja koulutusta. Dokumentointi on vapaasti käytettävissä ja on hyvin kattava.

Kuvassa 11 on näkymä OpenBravo-järjestelmän käyttöliittymästä.



Kuva 11. Näkymä OpenBravo-järjestelmän käyttöliittymästä.

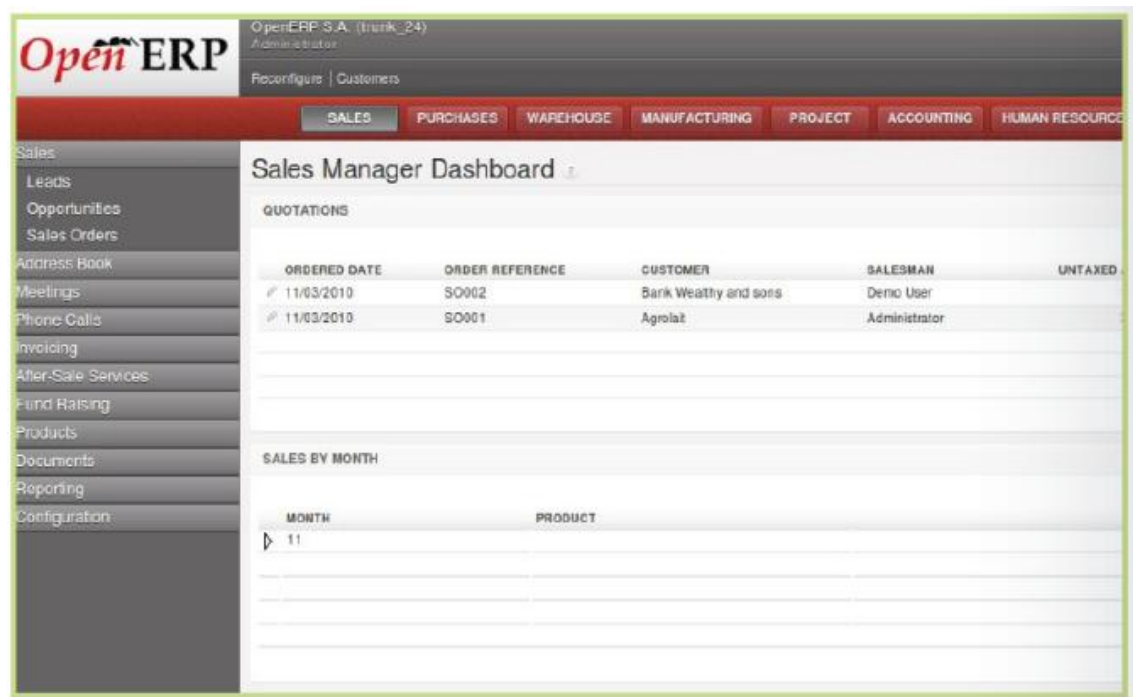
OpenERP <https://www.openerp.com/>

OpenERP-järjestelmää on kehitetty vuodesta 2005. Se on ohjelmoitu Python-ohjelmointikielellä ja toimii AGPL-lisenssin alaisuudessa. Järjestelmä toimii Linux- ja Windows-palvelimilla. Se tarjoaa myös client selainohjelman. Se tarjoaa myös tietojen tuonti ja vienti toiminnot. Ohjelmistolla on laaja kehittäjien ja tukijoiden joukko.

OpenERP-järjestelmän moduulit ovat asiakkuudenhallinta, ostojen hallinta, tuotannonohjaus, varastohallinta, projektinhallinta, laskentatoimen hallinta, henkilöstöhallinto ja myynnin hallinta.

Kehittäjät ja IT-palveluntarjoajat tarjoavat laajasti tukea. Käyttäjille järjestetään erimuotoista koulutusta. Dokumentointi on kattava, laadukas ja vapaasti käytettävissä.

Kuvassa 12 on näkymä OpenERP-järjestelmän käyttöliittymästä.



Kuva 12. Näkymä OpenERP-järjestelmän käyttöliittymästä.

[21]

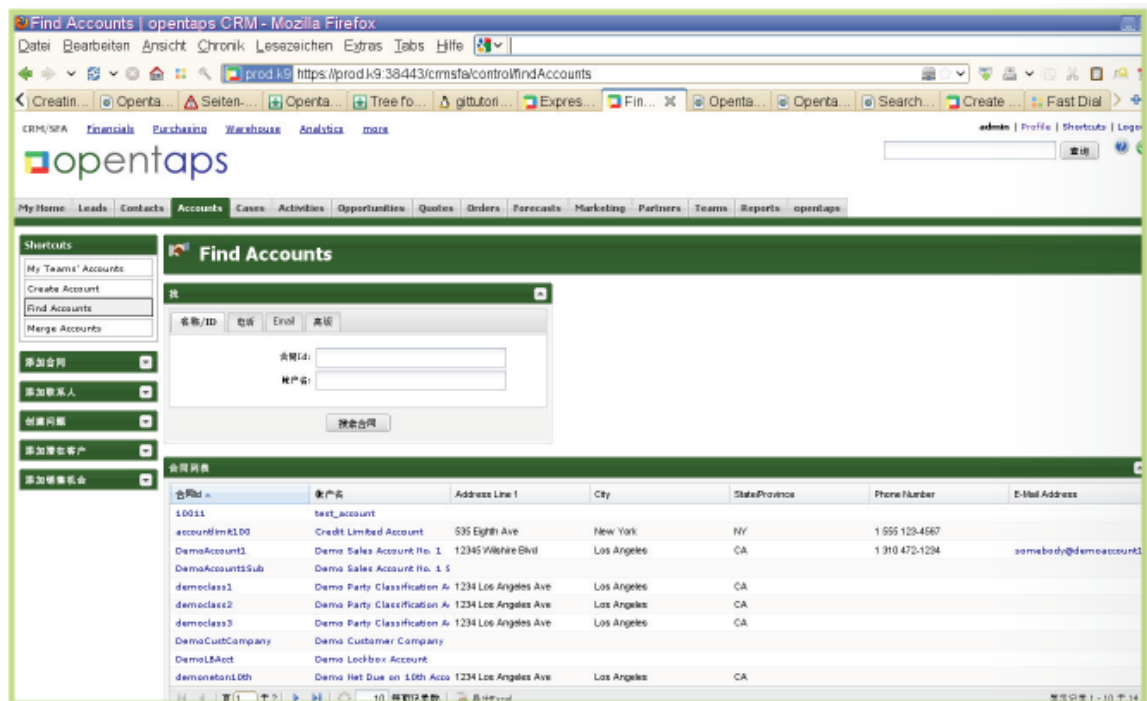
Opentaps <http://www.opentaps.org/>

Opentaps-järjestelmää on kehitetty vuodesta 2006 lähtien. Se on ohjelmoitu Java-ohjelmointikielellä ja toimii AGPL-lisenssin alaisuudessa. Opentaps-järjestelmä toimii Linux-, Windows- ja MacOS-käyttöjärjestelmillä. Ohjelmiston käyttöliittymä on selkeä ja käyttäjäystävällinen. Järjestelmää käyttävät suuryritykset kuten Honeywell ja Toyota. Opentaps on saatavilla myös pilvipalveluna ja mobiili-ERP-järjestelmänä. Se mahdollistaa myös tietojen tuonnin ja viennin. Ohjelmisto on suunniteltu pk-yrityksille. Verkkosivusto antaa erittäin hyvän kuvan eri toiminnoista.

Opentaps-järjestelmän moduulit ovat myynnin hallinta, laskentatoimen hallinta, raportointi, asiakkuudenhallinta, varastonhallinta ja tuotannonohjaus.

Laaja yhteisö tukee Opentaps-järjestelmää. Siihen kuuluvat seminaarit, koulutusta, räätälöintiä ja yrityksen vaativuusmäärittelyä. Lisäksi on tarjolla videoita, kirjoja ja verkkopalvelu.

Kuvassa 13 on näkymä Opentaps-järjestelmän käyttöliittymästä.



Kuva 13. Näkymä Opentaps-järjestelmän käyttöliittymästä.

[21]

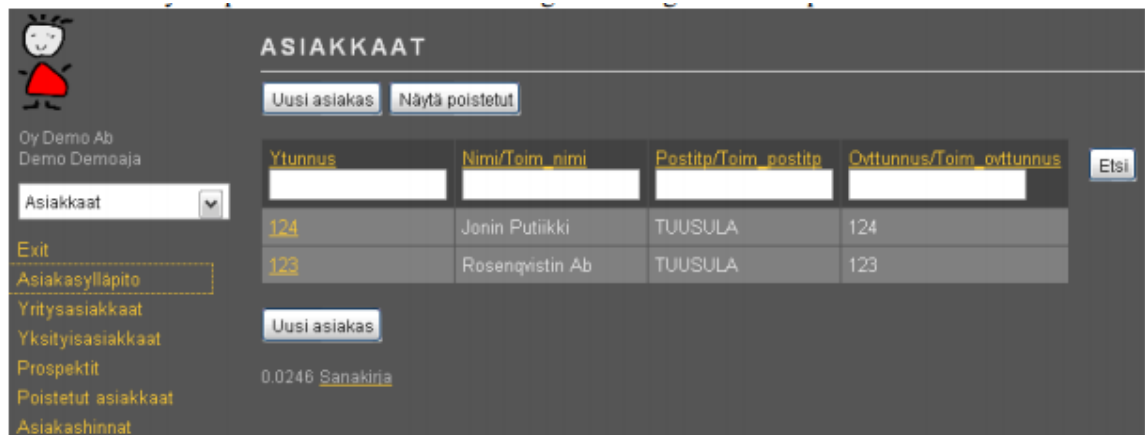
Pupesoftware <http://devlab.fi/#/home>

Pupesoftware-järjestelmää alettiin kehittää vuonna 2001 ja se on suomalainen. Se on ohjelmoitu PHP-ohjelmointikielellä ja julkaistu GPL-lisenssin alaisuudessa. Palvelin toimii Linux-käyttöjärjestelmällä. Pupesoftware-järjestelmä on monikielinen ja tukee useita valuuttoja. Ohjelmisto on suunniteltu pk-yrityksille.

Pupesoftware-järjestelmän moduulit ovat myynnin hallinta, ostojen hallinta, asiakkuudenhallinta, laskentatoimen hallinta, raportointi, valmistuksen hallinta ja varastonhallinta.

Järjestelmää tukee monipuolisesti suomalainen Develop Oy.

Kuvassa 14 on näkymä Pupesoftware-järjestelmän käyttöliittymästä.



Kuva 14. Näkymä Pupesoftware-järjestelmän käyttöliittymästä.

[26]

SQL Ledger <http://www.sql-ledger.com/>

SQL Ledger-järjestelmää on kehitetty vuodesta 1998 lähtien. Se on ohjelmoitu Perl-ohjelmointikielellä ja toimii GNU-lisenssin alaisuudessa. Se toimii Linux-, Windows- ja MacOS käyttöjärjestelmillä. SQL Ledger-järjestelmä on monikielinen ja tukee myös mobiilikäyttöä. Raportit voidaan suunnitella vastaamaan käyttäjien tarpeita. Järjestelmä on suunniteltu pk-yrityksille.

SQL Ledger-järjestelmän moduulit ovat myynnin hallinta, laskentatoimen hallinta, raportointi, asiakkuudenhallinta, hinnoittelun hallinta, varastohallinta, tilausten hallinta, ostojen hallinta, materiaalien hallinta ja vähittäismyynnin hallinta.

IT-palveluntarjoajat tarjoaa implementointia ja teknistä tukea. Lisäksi on tarjolla verkkopalvelu.

Kuvassa 15 on näkymä SQL Ledger-järjestelmän käyttöliittymästä.

The screenshot displays the 'Add Sales Invoice' window in the SQL Ledger application. On the left is a vertical sidebar menu with the following sections: 'Reconciliation Reports ...', 'Order Entry' (Sales Order, Purchase Order, Reports ...), 'General Ledger Reports', 'Goods & Services' (Add Part, Add Service, Add Assembly, Add Group, Stock Assembly, Reports, Parts, Services, Assemblies, Groups), 'Projects' (Add Project, Reports), and 'Reports' (Chart of Accounts, Trial Balance, Income Statement, Balance Sheet, Version). The main window title is 'Add Sales Invoice'. It contains several input fields: 'Record In' (1250-Accounts Receivables (Seattle)), 'Customer' (Good Customer-19101), 'Credit Limit' (0), 'Remaining' (-2,800), 'Currency' (USD), 'Ship via' (F.O.B. Somewhere), 'Invoice Number' (18382), 'Invoice Date' (07-01-), 'Due Date' (07-01-), and 'Order Number'. A table with columns 'Number', 'Description', 'Qty', 'Unit', 'Price', and '%' contains one row: '0001', 'LCD Monitor', '4', '1', '700.00'. Below the table are fields for 'No. 1', 'Delivery Date', 'Group wickets', and 'Project'. There are also fields for 'No. 2', 'Delivery Date', 'Group', and 'Project'. A 'Notes' section is present. At the bottom, there is a 'Payments' section with columns 'Date', 'Source', 'Amount', and 'Account'. The 'Account' field is set to '1010-Cash'. The footer includes navigation buttons: 'Invoice', 'Packing List', 'HTML', 'Postscript', 'PDF', 'Screen', and 'Group Iter'.

Kuva 15. Näkymä SQL Ledger-järjestelmän käyttöliittymästä.

[21]

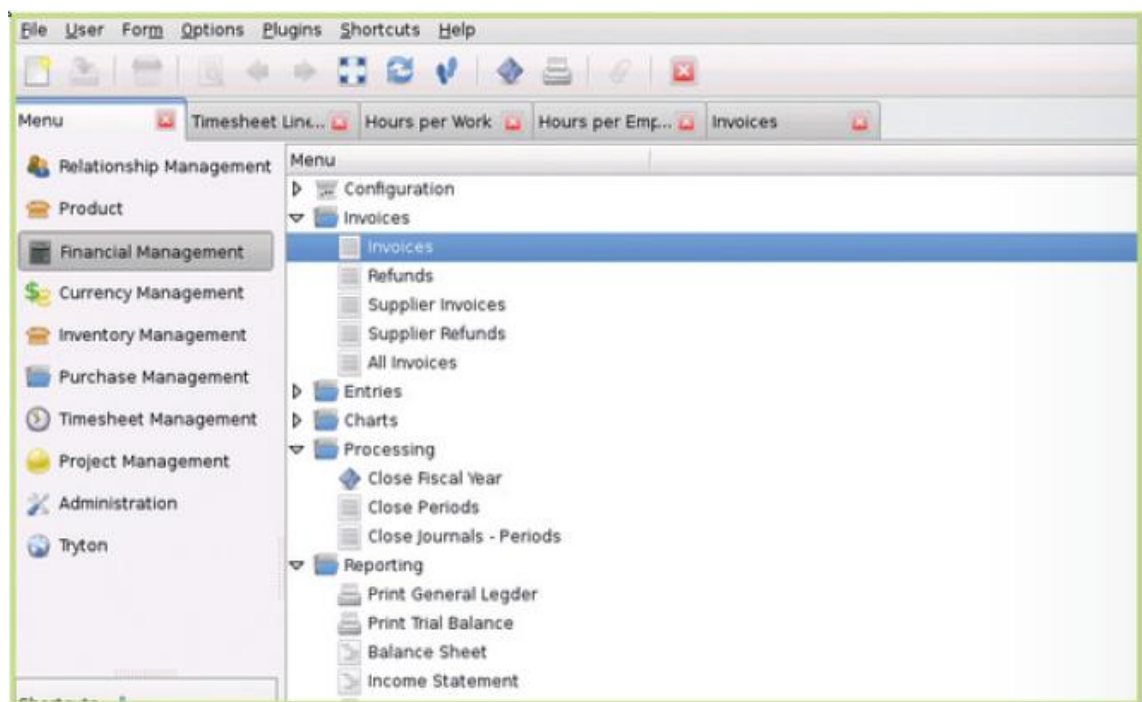
Tryton <http://www.tryton.org/>

Tryton-järjestelmää on kehitetty vuodesta 2008. Ohjelmisto on suunniteltu OpenERP-järjestelmän pohjalta. Se on ohjelmoitu Python-ohjelmointikielellä ja toimii GPL-lisenssin alaisuudessa. Järjestelmä toimii Linux-, Windows- ja MacOS-käyttöjärjestelmillä. Tietojen tuonti ei ole mahdollista. Vienti toiminto on kaupallisessa versiossa.

Tryton-järjestelmän moduulit ovat myynnin hallinta, ostojen hallinta, laskentatoimen hallinta ja varastohallinta.

Ohjelmiston kehittäjät ja IT-palveluntarjoajat tukevat ohjelmistoa. Siihen kuuluu implementointi, räätälöinti, seminaarit, koulutus ja ohjelmiston ylläpito. Dokumentaatio on korkeatasoista.

Kuvassa 16 on näkymä Tryton-järjestelmän käyttöliittymästä.



Kuva 16. Näkymä Tryton-järjestelmän käyttöliittymästä.

[21]

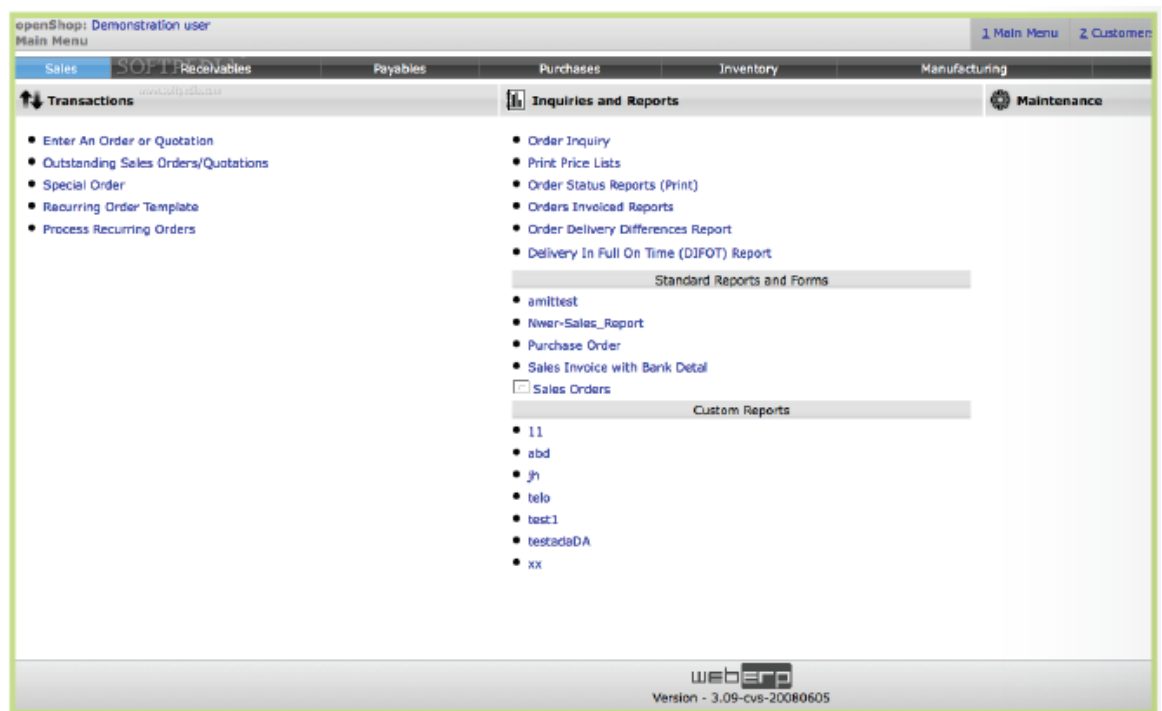
WebERP <http://www.weberp.org/>

WebERP-järjestelmää alettiin kehittää vuonna 2003. Palvelimen käyttöjärjestelmä on Linux. Se on ohjelmoitu PHP-ohjelmointikielellä ja julkaistu GPL-lisenssin alaisuudessa. Järjestelmä on monikielinen ja tukee useita valuuttoja. Verkkoyhteyksissä käytetään salausta. Se on myös suunniteltu pk-yrityksille.

WebERP-järjestelmän moduulit ovat myynnin hallinta, ostojen hallinta, laskentatoimen hallinta ja varastonhallinta.

Ohjelmiston yhteisö ja IT-palveluntarjoajat tukevat ohjelmistoa.

Kuvassa 17 on näkymä WebERP-järjestelmän käyttöliittymästä.



Kuva 17. Näkymä WebERP-järjestelmän käyttöliittymästä.

[21]

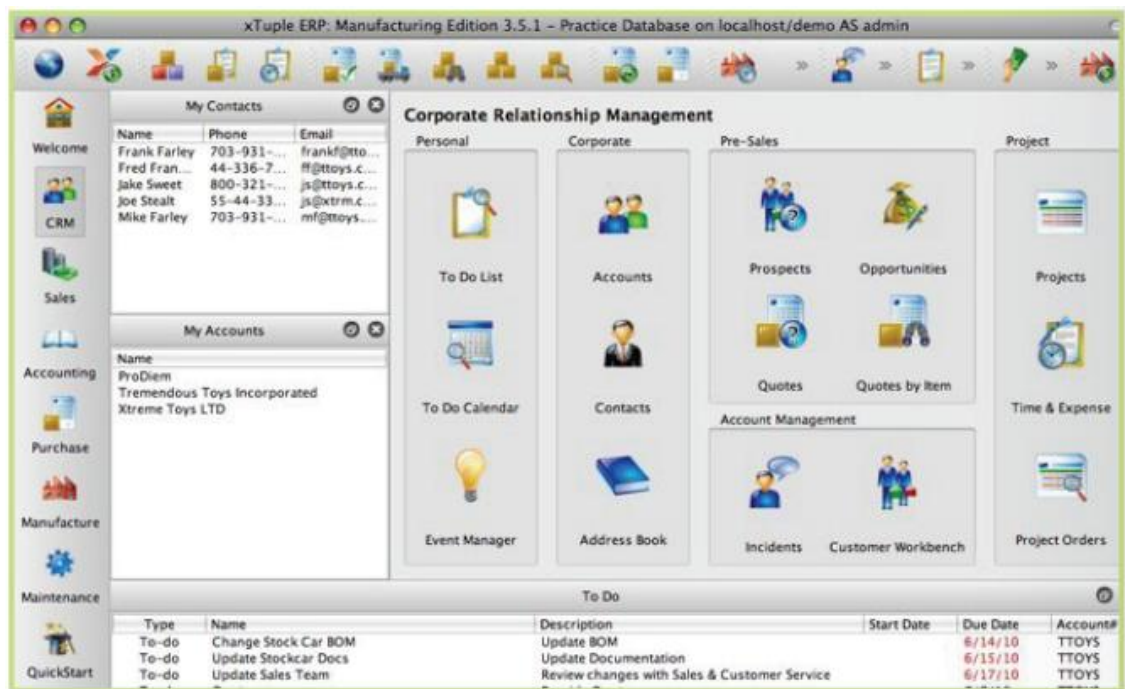
xTuple <http://www.xtuple.com/>

xTuple-järjestelmää alettiin kehittää vuonna 2001. Se on ohjelmoitu PHP-ohjelmointikielellä ja on julkaistu Common Public Attribution-lisenssin alaisuudessa. Se toimii MacOS-, Windows- ja Linux-käyttöjärjestelmillä. Käyttöliittymä on hyvin suunniteltu, selkeä, käyttäjäystävällinen ja on myös räätälöitävissä. xTuple tarjoaa myös tietojen tuonti ja vienti toiminnot. Verkkoyhteyksissä käytetään salausta. Se tukee useita valuuttoja.

xTuple-järjestelmän moduulit ovat tilausten hallinta, ostojen hallinta, tuotannonohjaus, materiaalien hallinta, asiakkuudenhallinta, tuotetiedonhallinta, myynnin hallinta, laskentatoimen hallinta ja raportointi.

Järjestelmän kehittäjät toimivat avoimesti ja tarjoavat hyvää yhteistyötä. Tukea on saatavilla monipuolisesti. Dokumentaatio on myös laadukas.

Kuvassa 18 on näkymä xTuple-järjestelmän käyttöliittymästä.



Kuva 18. Näkymä xTuple-järjestelmän käyttöliittymästä.

[21]

10 JOHTOPÄÄTÖKSET

Toiminnanohjausjärjestelmän hankinta on suuri investointi ja iso askel yritykselle, oli sitten kysymyksessä avoimen lähdekoodin tai suljetun toiminnanohjausjärjestelmän valinta. Sen implementointi saattaa kaataa koko yrityksen epäonnistuessaan. Tästä johtuen yrityksen kannattaa perehtyä aiheeseen hyvin. Yrityksellä tulee olla käytössään tarvittavat resurssit, avainhenkilöt ja osaaminen, jotta varmistetaan onnistuminen. ERP-järjestelmän konsultteja tulee käyttää, koska heillä hyvää tieto-taitoa. Tulee myös varmistaa, että konsultin tietotaitoa siirtyy yrityksen henkilölle mahdollisimman paljon ja dokumentaation taso on hyvä, jolloin saadaan sijoitukselle vastinetta. Koulutukseen tulee panostaa; jos järjestelmää ei osata hyödyntää, sen käyttö jää helposti vajaaksi. ERP-järjestelmä ei itsessään ratkaise ongelmia, mutta kun koko järjestelmän implementointi käydään läpi vaihe vaiheelta määrittelystä toteutukseen, tällöin myös yrityksen ongelmia pystytään varmasti ratkaisemaan.

Seuraavia asioita voidaan nostaa esiin, kun yritys miettii avoimen lähdekoodin ERP-järjestelmän hankintaa. Jos yrityksellä on käytössään omaa henkilöstöä, jotka ovat kiinnostuneita ja riittävän osaamisen omaavia henkilöitä niin avoimen lähdekoodin ERP-järjestelmä on hyvä vaihtoehto. Pitää kuitenkin varmistaa, että avoimen lähdekoodin ERP-järjestelmä ei jää vain joidenkin harvojen avainhenkilöiden käsiin. Jakamalla vastuuta taataan, että jos avainhenkilö ei ole enää käytettävissä niin pystytään paikkaamaan syntyvä aukko. Tämä asia pätee tietysti myös yleisesti tietojärjestelmiin. On huomattavaa, että avoimen lähdekoodin ERP-järjestelmää ei kannata valita pelkästään sen takia, että sitä voidaan sanoa ilmaiseksi, vaikkakin se on jo hyvä syy itsessään. Avoimen lähdekoodin ERP-järjestelmän implementointiin kuuluu muun muassa konsultointia, ylläpitoa ja koulutusta, joista muodostuu kustannuksia. Hyvän puolena on, että avoimen lähdekoodin ERP-järjestelmän kehitystyöhön osallistuu paljon henkilöitä eri puolilta maailmaa, joten se ei ole rajoittunut siihen, että toimittaja lopettaa toiminnan. Varteenotettava mahdollisuus on vain ladata verkosta soveltuva avoimen lähdekoodin ERP-järjestelmä ilmaiseksi ja asettaa tätä varten sopivat resurssit käyttöön. Avoimen lähdekoodin ERP-järjestelmän valinnassa kannattaa varmistaa, että siinä on sellainen lisenssi, joka mahdollistaa halutut ratkaisut, esimerkiksi jos halutaan aloittaa järjestelmän myynti muille.

Selvää on, että avoimen lähdekoodin ratkaisut tulevat näyttelemään suurempaa osaa myös avoimen lähdekoodin ERP-järjestelmissä tulevaisuudessa. Suomessa on vielä varsin vähän käytössä avoimen lähdekoodin ERP-järjestelmia. On varmasti olemassa

kielteinen ennakkoasenne avoimen lähdekoodin ERP-järjestelmää kohtaan, johtuen ainakin siitä, että siihen ei olla totuttu ja sitä ei pidetä toimivana näin isoissa tietojärjestelmissä. Sukupolven vaihtuessa avoimen lähdekoodin ERP-järjestelmä tulee varmasti näyttelemään suurempaa osaa, koska nuorempi sukupolvi on jo tottunut avoimen lähdekoodin ratkaisuihin, eikä heillä ole sellaista ennakkoasenteita, joita saattaa olla vanhemmalla sukupolvella.

Kun yritys valitsee sopivimman ratkaisun, ratkaisu jää itselle, koska kukin yritys on yksilöllinen eikä mitään selvää mallia ole olemassa. Liian laajaan tutkimukseen eri vaihtoehtoista ei kannata lähteä, koska se tulee myös hyvin kalliiksi. Tulee käyttää järkevä aika ja sitten tehdä tarvittavat päätökset ja lähteä määrätietoisesti toteuttamaan kyseistä valintaa.

LÄHTEET

- [1] Carol A. Ptak ja Eli Schragenheim. ERP Tools, Techniques, and Applications for Integrating the Supply Chain. 2. painos. CRC Press 2003. 464 sivua.
- [2] Jari Kettunen ja Magnus Simons (Toim.). Toiminnanohjausjärjestelmän käyttöönotto pk-yrityksessä. Teknologialähtöisestä ajattelusta kohti tiedon ja osaamisen hallintaa. Valtion teknillinen tutkimuskeskus, VTT Julkaisuja. Espoo 2001. 232 sivua.
- [3] Alexis Leon. Enterprise Resource Planning. 2. painos. Tata McGraw-Hill Education, 2008. 370 sivua.
- [4] Fiona Fui-Hoon Nah. Enterprise Resource Planning Solutions and Management. IGI Publishing 2002. 291 sivua.
- [5] T.H. Davenport, Putting the Enterprise into the Enterprise System. Harvard Business Review, 1998, 76(3). Sivut.121–131.
- [6] K. Kumar ja J. Van Hillsgersberg, ERP Experiences and Evolution. Communications of the ACM, 2000, 43(4). Sivut 23–26.
- [7] R. Tadjer, Enterprise Resource Planning. Internetweek, Manhasset, Apr 13, 1998.
- [8] D.E. O'Leary, (2000). Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk. Cambridge University Press, 2000, UK.
- [9] Jouko Karjalainen, Marja Blomqvist, Olli Suolanen. Kehittyvä toiminnanohjaus. Teknologiainfo Teknova 2001. 88 sivua.
- [10] Juhani Juuso ja Päivi Iskanius. Arviointikriteerit toiminnanohjausjärjestelmän valintaan, TOMI-raportti 5. Oulun yliopisto, Raahen toimintayksikkö, Raahen, 2009. 51 sivua.
- [11] Jose M. Esteves, Joan A. Pastor. An ERP Life-cycle-based Research Agenda. First International workshop in Enterprise Management and Resource Planning: Methods, Tools and Architectures – EMRPS'99, Venice, Italy 1999.
- [12] Fiona Fui-Hoon Nah ja Janet Lee-Shang ja Lau Jinghua Kuang. Critical factors for successful implementation of enterprise systems. University of Nebraska-Lincoln, Lincoln, Nebraska, USA, and University of Texas-Austin, Austin, Texas, USA. Sivut 285-296.

- [13] Anja Schatz, Peter Egri, Marcus Sauer. Open Source ERP, Reasonable tools for manufacturing. Fraunhofer Institute for Manufacturing Engineering and Automation IPA, 2011. 53 sivua.
- [14] David Bretthauer, Open source software: A history. Information Technology and Libraries, 2002. Sivut 3-10.
- [15] Chris DiBona, Sam Ockman, Mark Stone. Opensources : voices from the open source revolution. Sebastopol (CA) : O'Reilly , 1999. 272 sivua.
- [16] Kevin Carillo, Chitu Okoli. The open source movement: a revolution in software development. International Association for Computer Information Systems, 2008. 9 sivua.
- [17] Sauli Aalto-Setälä, Jari Leppäniemi. F/oss – vapaan avoimen lähdekoodin ohjelmistot, esiselvitys. Tampereen teknillinen yliopisto. Porin yksikkö. Pori 2005. 55 sivua.
- [18] JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta. Avoimen lähdekoodin ohjelmien käyttö julkisessa hallinnossa. JHS 169. Julkaistu: 23.02.2009. Versio: 1.1 5.10.2012. 34 sivua.
<http://docs.jhs-suositukset.fi/jhs-suositukset/JHS169/JHS169.pdf>
- [19] C. Boldyreff, K. Crowston, B. Lundell, A.I. Wasserman. Open Source Ecosystems: Diverse Communities Interacting. 5th IFIP WG 2.13 International Conference on Open Source Systems, OSS 2009, Skövde, Sweden, June 3-6, 2009, Proceedings Series: IFIP Advances in Information and Communication Technology, Vol. 299. 372 sivua.
- [20] James R. Borck. Best of Open Source Software Awards 2009. Infoworld. Infoworld Media Group. Aug 31, 2009.
- [21] Anja Schatz, Peter Egri, Marcus Sauer. Open Source ERP, Reasonable tools for manufacturing. Fraunhofer Institute for Manufacturing Engineering and Automation IPA, 2011. 53 sivua.
- [22] Nicolas Serrano, Jose Maria Sarriegi. Open Source Software ERPs: A New Alternative for an Old Need. IEEE Software. May/Jun 2006. Sivut 94-97.
- [23] Anonymous. xTuple; xTuple ERP 3.0 Wins 'Best Business Application' at LinuxWorld Conference & Expo 2008. NewsRx. Computer Weekly News. Aug 28, 2008. Sivu 127.

[24] Juho Eskeli, Samuli Heinonen, Tapio Matinmikko, Päivi Parviainen, Pasi Pussinen. Challenges and Alternative solutions for ERP's. VTT. Yrtti. Research Report VTT-R-05936-10. 55 sivua.